



University of HUDDERSFIELD

University of Huddersfield Repository

Beech, Guy

An Investigation of the Benefit of XML Technologies in Astronomical Data Interpretation

Original Citation

Beech, Guy (2015) An Investigation of the Benefit of XML Technologies in Astronomical Data Interpretation. Masters thesis, University of Huddersfield.

This version is available at <http://eprints.hud.ac.uk/28413/>

The University Repository is a digital collection of the research output of the University, available on Open Access. Copyright and Moral Rights for the items on this site are retained by the individual author and/or other copyright owners. Users may access full items free of charge; copies of full text items generally can be reproduced, displayed or performed and given to third parties in any format or medium for personal research or study, educational or not-for-profit purposes without prior permission or charge, provided:

- The authors, title and full bibliographic details is credited in any copy;
- A hyperlink and/or URL is included for the original metadata page; and
- The content is not changed in any way.

For more information, including our policy and submission procedure, please contact the Repository Team at: E.mailbox@hud.ac.uk.

<http://eprints.hud.ac.uk/>

An Investigation of the Benefit of XML Technologies in Astronomical Data
Interpretation

JAMES LAWRENCE GUY BEECH

A thesis submitted to the University of Huddersfield
in partial fulfilment of the requirements for
the degree of Masters by Research

The University of Huddersfield
School of Computing and Engineering
Centre for High Performance Intelligent Computing

October 2015

Copyright Statement

- i.** The author of this thesis (including any appendices and/or schedules to this thesis) owns any copyright in it (the “Copyright”) and s/he has given The University of Huddersfield the right to use such Copyright for any administrative, promotional, educational and/or teaching purposes.
- ii.** Copies of this thesis, either in full or in extracts, may be made only in accordance with the regulations of the University Library. Details of these regulations may be obtained from the Librarian. This page must form part of any such copies made.
- iii.** The ownership of any patents, designs, trademarks and any and all other intellectual property rights except for the Copyright (the “Intellectual Property Rights”) and any reproductions of copyright works, for example graphs and tables (“Reproductions”), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property Rights and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property Rights and/or Reproductions.

Abstract

This research work has demonstrated that from a variety of astronomical data sources and by using XML Technologies an advanced construct of usefully structured data can be produced in an XML Document. From this work an XML Schema has been developed in such a way that, in conjunction with the ROAD software also developed, it not only handles the variety of data used in this research through the use of XSLT transformation files but is also capable of being extended to include the full breadth of astronomical data storage content in the future, in all branches of astronomy. This research is in effect a proof of concept which is able to point the way towards the development of a W3C eligible XML Schema to be used as a standard in the astronomical research and industry sectors.

Acknowledgements

I would like to express my gratitude to my supervisor Professor Joan Lu and second supervisor Qiang Xu not only for encouraging me to carry out this investigation in the first place but also for their useful comments, remarks and engagement through the learning process of this Masters work. I also give thanks to Gill my partner for her encouragement and belief in me and all my family for their support and the confidence they had that I would see this research through to its end.

Table of Contents

Chapter 1 Introduction	7
Chapter 2 Literature Review	8
2.1. The History of recording Astronomical Data	8
2.2. Current problems in data retrieval	12
2.3. Solutions so far	13
2.3.1. OpenAstronomylog	13
2.3.2. NASA	14
2.3.3. AML – Astronomical Markup Language	14
2.3.4. RADAMS - a radio astronomical observation data model	14
2.3.5. Data Model for Astronomical DataSet Characterisation	15
2.3.6. The XML standards authority (W3C)	15
2.3.7. Astroinformatics	15
2.3.8. Summary of solutions so far	16
2.4. Advantages of XML based solutions	18
2.5. Publically available databases	19
2.6. In Summary	20
Chapter 3 Research Methodology – The Way Forward	21
3.1. The Theory behind this Research	21
3.2. Outlining the immediate goal	22
3.3. Methodology of the Proof of Concept	22
3.3.1. Define the requirements of the ROAD software system	23
3.3.2. The ROAD Software Requirements Specification	23
3.3.3. System Design	25
3.3.4. Implementation of the software system	25
3.3.5. Testing of the software system	25
3.4. XMLSchema Design Pattern	26
3.5. Data Sources	26
3.6. In summary	27
Chapter 4 Software System Development	28
4.1. Actual Data Sources	28
4.2. System Implementation	29
4.2.1. Inputs	31
4.2.2. Transformation	32
4.2.3. Outputs	33
4.2.4. Completed ROAD system	34
Chapter 5 Software System (ROAD) Runs	37
5.1. The purpose and planning of the System runs	38

5.2. Results of the System runs	40
Chapter 6 Data Analysis	42
6.1. Detailed Analysis of the results	42
Chapter 7 Discussion	49
7.1. Previous research	49
7.2. Summary of the main experimental results	50
7.3. Interpretation of the results	50
7.4. The significance of the results	51
7.4.1. How the data in the XMLDocument is held	52
7.4.2. How the XMLSchema handles the data	54
7.4.3. The importance of schema validation	56
7.5. Suggestions for improvements	57
7.6. The importance of the research	57
7.7. Barriers and limitations	58
7.8. Implications and Impacts	59
Chapter 8 Conclusions	61
8.1. Key Findings	61
8.2. Future Research	61
9 Appendices	63
10 References	119

Word Count: 16,037

List of Tables

Table 1: Table of branches of astronomy	10
Table 2: Moore's Law holds true in astronomy	11
Table 3: Expected growth in astronomy data volume	13
Table 4: Suitability of current schemas to achieve 100% coverage	16
Table 5: Complete Astronomy data coverage areas of the schemas	17
Table 6: XML Tools	18
Table 7: Proposed ROAD Actions	22
Table 8: Actual ROAD Classes	35
Table 9: Other ROAD Files	35
Table 10: Objectives of ROAD system runs	38
Table 11: Table of ROAD Runs	39
Table 12: Table of ROAD Run Results	41
Table 13: Table of Main Results	50
Table 14: Table of astronomy.xsd node use	53
Table 15: Suggested improvements to ROAD	57
Table 16: Table of datasource data types	58

List of Diagrams

Figure D.1: Research theory diagram	21
Figure D.2: Initial ROAD System Design	25
Figure D.3: ROAD System Use Case	29
Figure D.4: ROAD - Other Initial UML Diagrams	31
Figure D.5: Final ROAD System Design	34
Figure D.6: ROAD UML Class Diagram	36

Abbreviations

ROAD	Retrieval of Astronomical Data	PDF	Portable Document Format
CSV	Comma Separated Values	XML	Extensible Markup Language
BLOB	Binary Large Object	DTD	Document Type Definition
NASA	National Aeronautics Space Administration	XSLT	Extensible Stylesheet Language Transformations
FITS	Flexible Image Transport System	HTML	Hypertext Markup Language
HDF	Hierarchical Data Format	XSL-FO	XSL Formatting Objects
HIP	Hipparcos Catalogue	CDATA	Character Data

Appendices

Appendix 1 Astronomy.xsd structure	63
Appendix 2 astronomy.xsd	65
Appendix 3 test run screenshots	69
Appendix 4 The validated XMLDocument - example	74
Appendix 5 Web page produced in Run 1	83
Appendix 6 PDF produced in Run 1	84
Appendix 7 Web page produced in Run 2	85
Appendix 8 PDF produced in Run 3	86
Appendix 9 road.aspx.cs	87
Appendix 10 Query.cs	90
Appendix 11 QueryTransform.cs	94
Appendix 12 OutputTransform.cs	97
Appendix 13 mastQuery.xslt	102
Appendix 14 oecQuery.xslt	104
Appendix 15 simbadQuery.xslt	106
Appendix 16 htmlOutput.xslt	108
Appendix 17 pdfOutput.xslt	111
Appendix 18 Public databases	116
Appendix 19 - The complete RADAMS data model	116
Appendix 20 - Table of Design Patterns	117
Appendix 21 - Table of ROAD Run Actions	118

Chapter 1

Introduction

The current retrieval processes of existing astronomical records have a number of difficulties which, if they can be solved, would result in more reliable and cost effective acquisition of useful and valuable data. At the moment there is a vast amount of astronomical data stored in varying ways and formats, particularly the more historic records which quite often were limited in storage space, string length and other restrictions of technology at the time the data was saved. (NASA, 2014) Also to be considered is the increasing rate of astronomical data capture both by institutions and amateurs and also the growing different types of astronomical data collectors. (VAO, 2015) Although there are many institutions and individuals using good standards of astronomical data capture, are there any widely adopted standards of astronomical data recording that are currently in use? The same problem of varied data storage formats applies for methods of historic astronomical data retrieval because, of course, the processes of retrieval are very dependent upon the way data has been stored.

This means that the retrieval of astronomical data can require numerous steps. These steps typically involve different transformations with a number of (expensive) programmers. During these processes data could be inadvertently lost. Even more frequently the metadata associated with data can have some information missing or perhaps not bothered with at all. (NASA, 2014)

The purpose of the research work contained in this report then is to:

- Search for any existing widely used standards for astronomical data retrieval and their use.
- Identify a precise way forward to achieving improved efficiency of data retrieval, through gaining understanding of current literature, developing a new retrieval process and working towards an astronomical data structure standard.
- Identify and use astronomical data sources in a limited ‘proof of concept’ practical demonstration of a proposed solution to this data problem.
- Show how the research work contained in this report demonstrates such an improvement.

Following on from this introduction the structure of this report continues with a literature review in Chapter Two. This reviews previous documented work by others who have designed

astronomical data schemas and looks at the need for a comprehensive astronomical data structure. Chapter Three begins with a description of the theory driving the research in this report and then goes on to show the approach used in this research to provide a simple working example that enforces a common structure when carrying out astronomical data retrieval, in the form of a proof of concept design. Chapter Four shows how the Retrieval Of Astronomical Data (ROAD) software used in this research was designed and implemented whilst Chapter Five describes the test runs that were used and why these test runs were the ones that were chosen. In Chapter Six the results returned from the system software test runs are analysed. A general discussion of the findings of this research is carried out in Chapter Seven and following on from that, in Chapter Eight, conclusions are drawn and the possibilities of future research are considered.

These chapters are followed by the appendices and references.

Chapter 2

A Literature Review of Astronomical Data Storage and Retrieval

2.1. The History of recording Astronomical Data

It is worth noting that to search for the available literature use was made of the library facility ‘Summon’ at Huddersfield University. General searching of the internet was also carried out, mainly through Google, for journals, books and published papers.

Astronomical records have been kept in one form or another from antiquity but for the purposes of this study the scope of the records searched for has been limited to electronic records, although this does include older records that are now being saved in electronic formats, for example the Harvard College Observatory is currently digitizing its famed collection of glass plates (Hall, 2015) .

Originally astronomical observations were done with the naked eye, subsequently followed by optical instruments. From the nineteenth century onwards other forms of light that up until then were unknown began to be discovered, such as Infra-Red, Ultra-Violet and Gamma-Ray radiations. These days there is a good range of types of observations that are possible using the various detection specialties (Observational astronomy, 2015).

All of these observational types reside within the broad astronomy branch of ‘Observational Astronomy’. There are also currently another five primary branches of astronomy, making six in total (Outline of astronomy, 2015).

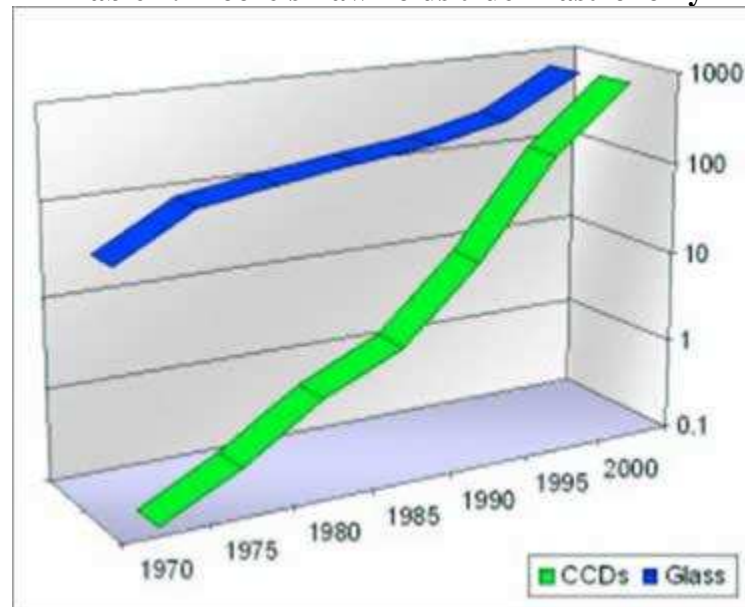
The location of observational detection types within astronomy as a whole is shown in table 1 below.

Table 1: Table of branches of astronomy

Primary branches of Astronomy	Sub branches
Archeoastronomy	
Astrobiology	
Astrochemistry	
Astrophysics	
Observational Astronomy	Gravity Wave Detector Infra-Red imaging telescope Microwave telescope Neutrino Detectors Radio telescope Short wavelength detector (Gamma, Ultra-Violet) Submillimetre Visible spectrum imaging telescope
Planetary Science	

There are of course within the other primary branches of astronomy more sub branches that are not shown here as they were not developed during this research.

As recently as the 1990's virtually all astronomical records were kept in the physical forms of paper, microfilm and reels of magnetic tape, rather than in digital form. However by 2007 the vast majority of all of the world's technological memory was held in digital format. In fact, "in 1986 only 1% of the world's capacity to store information was in digital format, which grew to 3% by 1993, 25% in the year 2000, and exploded to 97% of the world's storage capacity by 2007" (Lopez, 2014). Astronomical records are no exception from this step change in data saving technology. At the same time that the vast majority of existing records were being saved onto electronic databases the quantity of astronomical data being recorded was rapidly increasing, due partly to the needs of big data storage by the increasing number of digital sky surveys but also due to the increasing number of observational facilities available, both in the professional and amateur range of observers. Also the growth in the amount of data capture by digital recording devices on telescopes further increases the ability to capture information. (VAO, 2015)

Table 2: Moore's Law holds true in astronomy

Blue curve is total area of astronomical telescopes (glass), in square meters. Green curve is growth of CCDs, in billions of pixels. The amount of data being collected (green line) is following Moore's Law. (VAO, 2015)

It can be seen from the table above that the rate of increase of data collection by astronomical telescopes is currently exponential, due to the fact that the amount of data stored has been doubling on each previous year for a number of years now. The Large Synoptic Survey Telescope currently under construction is an example of the big data collection capacity of modern astronomical observation projects. (Stuart, 2015) In addition astronomical data is being saved to databases in numerous formats with no centrally agreed standards. (Pepe, 2015)

Retrieval is carried out in many different formats also. As data has no value unless it can be retrieved, it follows that the more easily retrievable the data, the more the data is worth both financially and scientifically. The motivation for better retrieval is a clear one. However the accelerating rate of astronomical data capture raises the question of how best to save and use it. (Andersen, 2014)

2.2. Current problems in data retrieval

The problems were summed up most succinctly by a group working at NASA's Astronomical Data Centre (ADC Archive) from around the turn of the century that was subsequently dissolved in 2002. This group concluded that the three key problem areas were what they called the 'multiplicity problem', the 'transformations problem' and the 'archaic problem'. (NASA, 2014) They argued that the multiplicity of data formats were created using 'home grown' solutions, quite often based on 1970's technology. They noted that even when there was an attempt to adopt standard formats, subclasses of incompatible formats were still formed and used. An example this team gave of transformation problems was that of satellite data which are commonly converted five or six times before reaching end users due to the number of incompatible format types in use. For example: from telemetry to science format, then to analysis software formats such as FITS, HDF or others. Then perhaps to PDF for publication, FITS again at data centres and then to GIF or MPEG for public consumption. Even if it was included at the beginning most metadata is lost during this processing. (NASA, 2014)

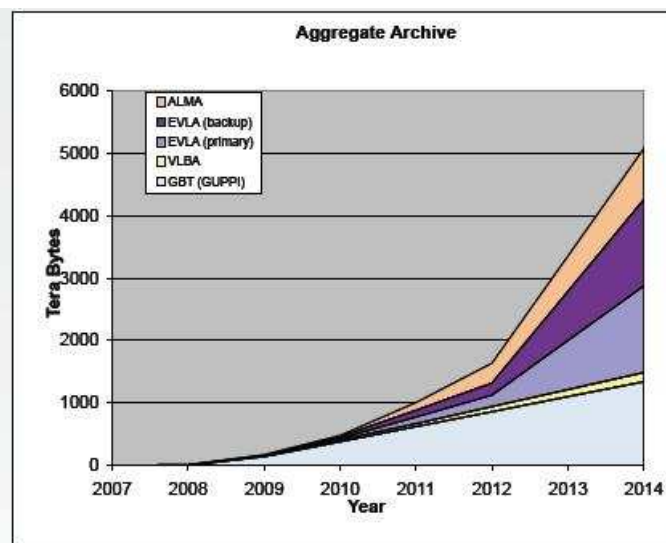
Finally the team looked at archaic problems where vintage data has its own set of problems which may include:

'restricted number of characters for variable names; restricted record length; restricted number of records; lack of hierarchical structure; inability to directly query multiple files; lack of pointers or referencing mechanism; discipline specific attributes; difficulty in extensibility or interoperability; insufficient self-description; no hyperlinking; not easily Web-enabled'. (NASA, 2014)

Other difficulties have been identified by SkyQuery who recognise that due to the fuzzy nature of images at the extreme range of detectors, data from separate telescopes needs to be combined to get the most complete data. So that we are talking about the retrieval of multiple data sets to produce a single good quality result. (SkyQuery: Federating Astronomy Archives, 2014) This problem is echoed by Csillaghy, Hinterberger and Benz who also point out that astronomical data can be noisy and diffuse. (Content-Based Image Retrieval in Astronomy, 2014) This need to combine data from several sources to attain a high quality data set describes yet another problem in manipulating astronomical data that would be helped by improved retrieval of both data and metadata.

Already by 2010 it was being felt that the vast amounts of astronomical data being captured meant that Astronomers were starting to move to a more passive role, letting the telescopes themselves choose when an item of interest was detected and then saving the data automatically. In Space.com magazine this was commented on with the view that the bottleneck of astronomical data was shifting from discovery to data management. (Wall, 2014) The table below shows the aggregate growth of the volume of astronomical data that has been stored.

Table 3: Growth in astronomy data volume (Berriman, 2015)



So, in addition to those of multiplicity, transformations and archaic it is clear that there are also the problems of ‘quantity’ and ‘multiple sources’. This review looked to see firstly what solutions (if any) have been attempted and also whether they have been widely used.

2.3. Solutions so far

A number of organisations have made a start on devising XML schemas to regulate the structure of stored astronomical data. These are:

2.3.1. OpenAstronomylog which has a schema for astronomical observations, although only in the visual range. This is really quite comprehensive in providing the attributes and nodes to contain records of the detail of observations. However there is no place to record the location of any image that this information may refer to. Also there appears to have been no activity on the site since 2010. Their claim is that they have produced a free and open XML Schema definition

for all types of astronomical observations. (Openastronomylog, 2014) There has been more work done on the detail of this schema than any others that were found and it is useful in showing the spread of information about observations that needs to be recorded. It is also pointing in the right direction ethically as the license conditions enable free use of the schema, but it is far from complete. Any comprehensive schema would need to be able to encompass all primary branches of astronomy, not just observational astronomy in the optical range.

OpenAstronomylog's schema with its 'Base' root node of optical observation prevents any other observation types other than optical observation from being included. Therefore this is not a design that can be extended to include primary branches of astronomy, it would have to be redesigned completely. This is a 'non extensible' design.

2.3.2. NASA is another organisation that made an attempt at producing a schema. They ran a research project to find a way to convert their Astronomical Data Centre archive to XML to improve the archive and searching processes (NASA, 2014). They proposed and began developing an XML that they named XDF, but the plug was pulled on the funding in 2002 and the project and team disbanded without achieving any kind of product. This is very disappointing given the organisation's huge resources. Indeed the acronym XDF is no longer used by them for this purpose, but is now re-used by NASA to refer to the deep space images of very remote galaxies captured by their Hubble Telescope (XDF – eXtreme Deep Field) (NASA Science, 2014). A DTD file was produced by the project for structuring observation data in the visible range and can still be found at: <http://xml.coverpages.org/FITSML-200103dtd.txt> (NASA, 2015). Again this is a limited design that is 'non extensible'.

2.3.3. AML – Astronomical Markup Language This is a DTD file last updated on 15th May 2000.

This was a precursor to the possible development of a schema that has so far not been developed. (Guillaume, 2015). The DTD holds a structure for an observation including information about the observation target, the telescope used and the observer. There is no other structure in the DTD. Again, a design that is limited and 'non extensible'.

2.3.4. RADAMS - a radio astronomical observation data model.

The root class is observation, with a structure for individual observation detail.

This is a comprehensively developed schema model (See Appendix 19), with further development on going in that area, specifically radio interferometry, but by setting the root effectively at individual radio observations, the schema is restricted to that single area of astronomical observation data type. (Santander-Vela, et al., 2012). For this reason, it is yet another ‘non extensible’ design.

2.3.5. Data Model for Astronomical DataSet Characterisation – International Virtual Astronomy Alliance (IVOA).

This is an abstract structure for astronomical data definition designed to be used for many types of astronomical data sets. It is well defined and detailed for individual observation data, though it has no wider astronomical structure and would need to be inserted into a wider schema. Last developed in May 22nd 2007, there is no evidence that it has been taken up for use. Although it could fit within a wider astronomical XMLSchema there is no evidence of this being done as yet. (IVOA Data Model Working Group, 2015). This is another ‘non extensible’ design.

IVOA are making a concerted effort to produce specific schemas for astronomical data, by producing specific schemas for specific areas, such as the dataset characterisation schema mentioned above. IVOA has not produced any schema extensible to the full width of astronomical data. (International Virtual Observatory Alliance, 2015)

2.3.6. The XML standards authority (W3C) has nothing specifically relating to astronomy in all of its history, from the beginnings of W3C in 1996 up to the present day. (W3c Standards, 2014) If anybody managed to produce a high standard all-encompassing XML astronomical schema, one would hope and expect it to turn up here. This is the place such a schema would need to be declared if it were to stand a good chance of being widely accepted.

2.3.7. Astroinformatics This is an approach that is not XML centred.

This is part of the new science of X-informatics. It is the analysis and interpretation of data to make new discoveries in astronomy by developing formal sub-disciplines that are information-rich and data-based. Disciplines of a similar approach include bioinformatics and geoinformatics. (Borne, 2014)

As this article points out astronomers are increasingly finding that their first hand experiences with observations are through databases rather than taking a trip to an observatory, simply due to the increasing amount of captured observation data that is yet to be analysed. The ability to meaningfully query large distributed repositories and thereby make data-driven new discoveries is becoming ever more important. Rather than an XML schema solution, in this research article the author Borne envisages the creation of an ‘artificial genome’. (Borne, 2014) This idea has been ‘borrowed’ from the field of bioinformatics genomics research. This genome will lead to ‘genes’ that represent classes of astronomical objects. To this end an Astroinformatics research program is proposed.

This article is from 2010 and although it is claimed that NASA is looking at the idea, nothing concrete has yet emerged.

2.3.8. Summary of solutions so far

Given that there are eight main types of detectors that record observational astronomy data the current schemas found by this review, which all concentrate on observational data, can be summarised by the number of these areas catered for, their extensibility to encompass other areas of detection data and whether development work is on-going with the schema design. This information is shown in table 4 below. Note that each detection data type is represented as a percentage of the whole of the ‘Observational Astronomy’ data area (12.5%). Observational Astronomy itself is of course, as previously noted above, only one of six main areas of astronomy.

Table 4: Suitability of current schemas to achieve 100% coverage of the Observational Astronomy data areas

Name	Latest change date	Work on-going	Completed development	% of current data range	% of possible future data range	Suitability to achieve 100%
OpenAstronomylog	Aug 2011	✓	✗	12.5	12.5	✗
Astronomical Markup	May 2000	✗	✓	< 12.5	< 12.5	✗
RADAMS	May 2012	✓	✗	12.5	12.5	✗
IVOA	May 2007	✓	✗	37.5	Unknown	✗
XDF	Nov 2000	✗	✗	12.5	12.5	✗
XML Standards Authority	April 2015	✗	✗	0	0	✗
Schema created during this research	April 2015	✓	✗	25	25	✓

Table 4 above shows that of these solutions about half are ongoing and only one has completed development. None can be extended across all primary branches of astronomy with the exception of the schema included at the bottom of the table which is the schema developed during this research.

Table 5: Complete Astronomy data coverage areas of the schemas (current and possible)

Schema or DTD	<----- Observational ----->								<-- All ---->	
	Visible	I-Red	Radio	Mwave	Gwave	Shortwave	Neutrino	Sub-mm	Extensible in Observational Astronomy sub branch	Extensible across all primary branches
OpenAstronomylog (Schema)	✓								X	X
Astronomical Markup (DTD)	✓								X	X
RADAMS (Schema)			✓						X	X
IVOA (Schema)	✓	✓	✓						X	X
XDF - NASA (DTD)	✓								X	X
XML Standards Authority									X	X
Schema created from this research	✓	✓							✓	✓

Table 5 above shows the same solutions as table 4 but concentrates on their current schema coverage, the possibility of extending it across the ‘Observational Astronomy’ sub branch of astronomy and also any possible extensibility across all of the primary branches of astronomy.

From this table it is clearly shown that all schemas that have been found during this research are limited in their scope. They only cover a few of the detector types in the Observational Astronomy sub branch. This is equally true of the schema created during this research, shown in the bottom row of the table. However, it is also stated in table 5 above that only the schema developed during this research is capable of full extension, the reasoning for this is to be found in section 7.4.2 of this document.

There does appear to be awareness of the problems caused by a lack of common, widely accepted standards, as can be seen from the range of articles written on this subject. (Google Scholar, 2015) Some limited schemas have been produced and made public but as can be seen from the tables above there is no evidence of a common purpose amongst the astronomical community to create a widespread and acceptable information structure when presenting astronomical data that is able to encompass all of astronomy. To create such a schema and get it accepted, would enable

the retrieval of huge amounts of complete and accurate astronomical data in a more cost effective and reliable way.

2.4. Advantages of XML based solutions

There are a number of technical tools available for creating an all-encompassing structure for astronomical data retrieval.

It is possible that AstroInformatics may enable such a structure to be created. The idea of creating a solution around the concept of an artificial genome is an interesting one. However it is unclear how this can be done, whereas it is possible to visualise an XML based solution. The restriction of time during this Masters level research also dictates going with the quicker and more productive route of an XML solution. Therefore the decision taken was to proceed with an Extensible Markup Language (XML) based solution. XML has a number of powerful tools, the following of which have been determined as being of relevance (XML Technology - Schema, 2014):

Table 6: XML Tools

Xml Document	A file containing information in a clear markup, node type structure. It is of a clearly readable form - it is both machine and human readable
Transformation – XSLT	An XSLT file is used to transform xml documents into other structured file types such as HTML for web pages, XSLFO to produce PDF, or just differently structured xml documents for machine to machine information transfers.
Schema	A schema is an xml file that contains a set of rules to describe an xml structure. This schema can be associated with an xml document. When this is done, the schema ensures that the xml document can only be created along the lines of the schema rules. So this is a way of ensuring conformity of data structures when xml documents are produced. Of all of them this tool is the crux of achieving improved data retrieval.
X- Path	A tool enabling navigation through an xml document.
XML Query	This uses XQuery which is able to retrieve data from relational databases and from documents on the web.
Presentation – XMLFO	Stands for XML Formatted Object. It enables the presentation of data in xml documents in various formats such as PDF or RTF files.
XML Encryption	This provides the ability to encrypt and decrypt all or part of xml documents to ensure the security of the data that they contain.

These tools have been chosen as they could be pulled together to provide a way that would take disparate data and restructure it whilst ensuring that the requirements of a common data format, metadata retention, data security and valuable useful output (quality assurance) could be satisfied.

The design of the XMLSchema was going to be absolutely crucial to enable a successful outcome to this research. XMLSchema design patterns were studied to see which ones would be most appropriate to the research. There are a number of clearly defined pattern types. A description of these types had been sourced from Oracle online (Sum, 2014) as a table of design patterns that can be seen in Appendix 20.

What was needed was a schema design with a single root element. It was from this root that all other types and elements were derived. A lot of the elements are of complex type and also a lot of the elements are able to be used multiple times. Having looked at the information about the patterns in the table, the pattern that best fitted these requirements was that of the 'Venetian Blind'. The XML standards authority site points out that this pattern simplifies mapping to and from a data model. (XML Schema Patterns for Common Data Structures, 2014)

2.5. Publically available databases

These databases shown below are large, extensive and were found to be easily accessible for querying online. It means that they would be useful as a source for the proof of concept ROAD application that would be used during this research.

[Mikulski Archive for Space Telescopes \(MAST\)](#) (Mikulski Archive for Space Telescopes (MAST), 2015)

[Strasbourg astronomical Data Center - SIMBAD](#) (Strasbourg astronomical Data Center – SIMBAD, 2015)

[Open Exoplanet Catalogue \(OEC\)](#) (Open Exoplanet Catalogue, 2015)

For details see Appendix 18.

2.6. Summary of this literature review

It has been found that the great majority of astronomical records in all their wide variety right back to ancient times are now held in digital format in electronic databases, in a wide variety of formats. There has been in recent years an extremely rapid increase in the rate at which new data is being saved and this is a trend that is going to carry on into the future.

Due to the variety of the record types, the data structures and the sheer volume of data the retrieval of astronomical data can be complicated, expensive and incomplete. There is no widely used, comprehensive standard of data recording used in astronomy. Some proposals have been suggested and a few attempts have been started but schemas produced so far have been limited in scope in that they have been designed to cover designated sections of astronomy and not astronomy as a whole. A schema that is capable of providing data structure to the whole of astronomy will take time to develop. Therefore it is essential that in the initial stages care is taken to ensure that the structure enables later additions to it and will not require changes or otherwise redesigning of any existing schema nodes. Child nodes should only be added in a way that preserves and enhances the existing schema.

It was concluded that XML Technologies could provide the means to create a practical workable solution, but was yet to be shown in practice. Therefore this research was designed to both create a workable, expansible XMLSchema and develop software to demonstrate this schema in use, as a proof of concept.

Chapter 3

Research Methodology - The Way Forward

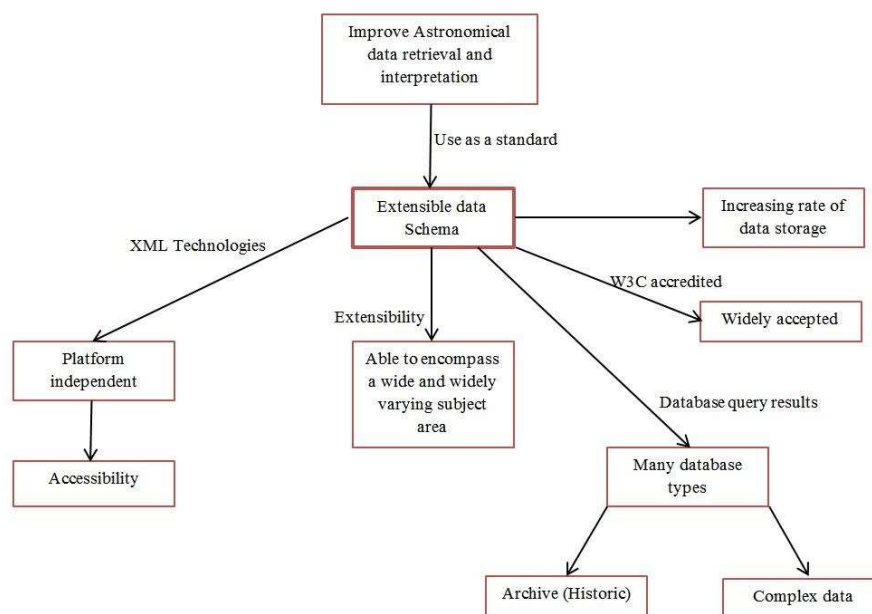
The literature review showed that although some astronomical database schemas had been developed, they were all designed around areas of data within astronomy as a whole and not meant to cater for all of the astronomy subject areas. In addition these developed schemas were designed in a way that meant they were not extensible and therefore could not be added to in any way that would enable the formation of one schema that catered for all of astronomy data. Therefore it was apparent that a new schema needed to be written. This schema should have a structure that could be useable for all of astronomical data, across all of its primary branches.

In addition software was to be developed to demonstrate the usefulness of this new schema in providing a standard structure for data retrieved from existing astronomical databases. This software was given the name 'Retrieval of Astronomical Data (ROAD)'.

3.1. The Theory behind this Research: How to improve astronomical data retrieval

A Conceptual Diagram is used below to show relationships between key factors derived from the literature review and the central construct of this theory which is the 'Extensible data schema':

Figure D.1: Research theory diagram



This extensible schema acts as a controller ensuring correct storage structure when astronomical data is retrieved. XML technology is platform independent ensuring that this schema is also platform independent. It will be able to accommodate all data types through conversion using XSLT technology. W3C accreditation would help this schema to become widely accepted and the extensibility of this schema ensures that it would be able to develop to encompass the wide subject area of all the primary branches of astronomy. This extensible schema would be used with a custom software application developed as part of this research to control the structure and content of astronomical data retrieved by that custom application from existing astronomical databases.

3.2. Outlining the immediate goal

The immediate goal of this research was to develop the custom application ROAD along with the extensible XMLSchema to demonstrate the practicalities of the proposed astronomical data processing. This proof of concept involved the creation of a piece of working software that is capable, through the use of XML Technologies, of a number of actions listed in Table 7 below:

Table 7: Proposed ROAD Actions

Querying several kinds of astronomical data sources to produce a single XML Document
Being controlled by an XML Schema
Carrying out transforms using XSLT files
Creating XSLFO documents to produce PDF, HTML and machine readable documents that will display and use the information in the XML Document in different ways
Showing an 'end to end' example of the data management that has been discussed
Using security through encryption to be selective of data exposed

It was intended that the astronomical Schema astronomy.xsd would only be developed far enough to enable this demonstration. However the Schema was also to be structured in such a way as to enable its future expansion to encompass all astronomical data recording across all primary branches of astronomy.

3.3. Methodology of the Proof of Concept

The methodology is outlined below as follows.

3.3.1. Define the requirements of the ROAD software system

Working environment:

The ROAD software is written in C# and ASP.net using Microsoft Visual Studio IDE. This working environment was chosen as it was familiar as well as the fact that it is a very capable environment for XML work.

Inputs:

The data was retrieved from several databases holding astronomical data, the database types to include a large data retrieval system, a relational database and one other type. This required the creation of suitable queries and the results combined into a single query results object.

Expected Outputs:

The primary research output was that of an XMLSchema file capable of managing the production of the proof of concept XMLDocument but is also extensible to the full breadth of astronomical data across the disciplines of Astronomy. Also produced were XSLT files for the transformation processes to create the XMLDocument from the query results object and then, from the XMLDocument, the XSLFO files to produce the system outputs of PDF, HTML and machine readable files.

It is now possible to produce a software requirements specification for the ROAD application.

3.3.2. The ROAD Software Requirements Specification

Introduction

The software will query selected astronomical databases, transform the data into an XMLDocument of standard structure controlled by an XMLSchema and produce useable outputs of varying chosen types.

Purpose

The purpose of this specification is to give all of the information necessary to design, develop and test the ROAD software.

Scope

This specification contains a complete description of the functionality of the ROAD application.

System Overview

The user first chooses the data sources and then the desired outputs. An astronomical object is then selected. The database connections are tested. If not all are connected the user is given the choice to abort or to carry on with those successfully connected. The system queries the databases, transforms the

returned results and puts them into a new XMLDocument. The XMLDocument is then used to produce the requested outputs. A log file is kept for all actions and errors.

Use Cases

Name	Choose data sources
Summary	Enables the user to select multiple databases
Rationale	So that data may be queried from multiple astronomical databases
Users	All Users
Preconditions	Awaiting User input
Basic Course of Events	User must select each database from a list
Alternative Paths	User can exit the ROAD application
Postconditions	Ready for the Users to choose from a list of available outputs

Name	Choose outputs
Summary	Enables the user to select multiple data output types
Rationale	So that data may be output in multiple formats
Users	All Users
Preconditions	Awaiting User input
Basic Course of Events	User must select each output type from a list of PDF, HTML and XML
Alternative Paths	User can exit the ROAD application
Postconditions	Ready for the Users to instruct ROAD to carry out processing

Name	Process data
Summary	Enables the user to begin processing
Rationale	So that astronomical data may be queried, transformed and saved
Users	All Users
Preconditions	Databases and data output types have been selected
Basic Course of Events	Data will be queried, the results transformed and saved to an XMLDocument. Output files will be created and saved. A log will be written to file.
Alternative Paths	User can exit the ROAD application or select other datasources or output types.
Postconditions	Processing run completed.

Functional Requirements

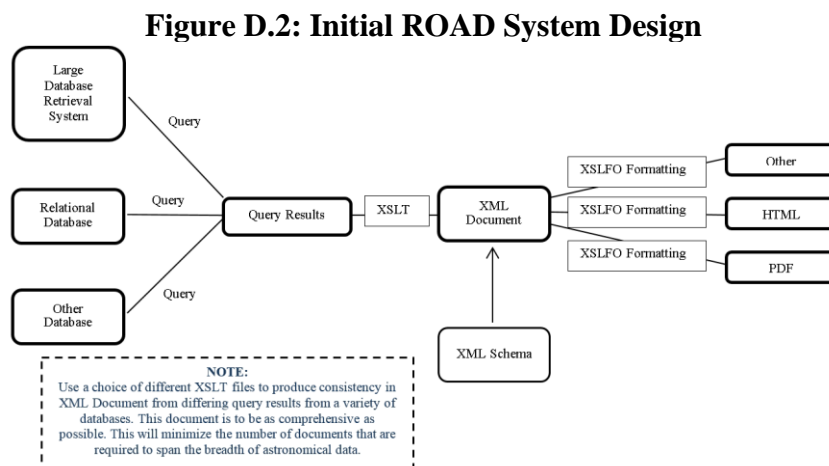
Chosen databases are to be queried and the retrieved data held in results objects. In turn each set of results is transformed using XSLT and inserted into an XMLDocument whose structure is defined by An XMLSchema document. This XMLDocument is then saved. This XMLDocument is validated against the XMLSchema document. If validation is successful output documents of the chosen types are created and saved to file. A log of all actions is written to file as are any errors including any validation failure.

Nonfunctional Requirements

The user interface is to be of simple design to minimize as far as possible development time.

3.3.3. System Design

The diagram shown below is a high level one. Lower level diagrams were produced during the implementation process in section 4.2 of this document.



3.3.4. Implementation of the software system

The system was designed from the diagram above using UML diagrams. Use Case, Sequence and Class diagrams were used in the first instance to produce lower level designs followed by the subsequent coding. For reference to this work see section 4.2. of this document.

Once completed, because of the Microsoft working environment, deployment of the software was easily made to any Windows machine.

3.3.5. Testing of the software system

Firstly the initial data queries needed to produce the expected query results object. Then to check that the Transform (XSLT) was performing as expected.

The XMLSchema had to be seen to be allowing creation of xml documents according to its specification. Equally, any error created from the incoming data being incompatible with the schema was to be properly stated and output to a log file.

Checks were to be made that the expected outputs of PDF, HTML and machine readable file were being produced.

Error testing was to be done. This was to maximise the information coming out of the system run.

3.4. XMLSchema Design Pattern

To ensure a fully expandable XMLSchema a pattern with a single global(root) element of ‘Astronomy’, reusable complex types and reusable element groups would be the best pattern to help create the single root based, extensible schema design that was sought after.

So the decision was to base the schema on the ‘Venetian Blind’ pattern. (Sum, 2014). The root to be ‘Astronomy’ which was given the primary branches of astronomy as child nodes as shown here:

- Observational Astronomy
- Astrophysics
- Astrobiology
- Astrochemistry
- Archeoastronomy
- Planetary Science

The focus of this research work was to develop the schema in the primary branch of ‘Observational Astronomy’ within which the secondary branches of astronomy were given the following child nodes:

- Visible
- Infrared
- Radio
- Microwave
- Gravity wave
- Shortwave
- Neutrino
- Submillimetre

3.5. Data Sources

To use a representative amount of disparate data it was decided to use three different sources.

Each source would be of a different type of database.

The purpose of using these different types was to demonstrate a capability of data retrieval that, although not comprehensive, is nonetheless from a wide range of data types.

A large database retrieval system

These are commonly image retrieval systems. The large image files are stored along with their associated metadata, for the purposes of search and identification.

A relational database

This type is one of the most commonly used in the storage of structured data in recent years.

An older type of database

The use of this type would demonstrate the ability to retrieve data from older storage systems.

At this point the exact databases to be used had not yet been precisely determined. However thanks to a contact provided through the University of Huddersfield a discussion was in progress with experts in the field of astronomy at the UK Astronomy Technology Centre (ATC) in Edinburgh about the provision of access to suitable data. This centre is part of the Science and Technology Facilities Council and is the national centre for astronomical technology in the UK. (UK Astronomy Technology Centre, 2014)

3.6. In Summary

The goal was to produce software that would demonstrate a working proof of concept model of the XMLSchema being put to use. The XMLSchema was to be developed far enough to handle the demands of the working model, whilst at the same time remaining extensible in the future to encompass the complete width of the primary branches of astronomical data.

The method chosen to demonstrate the XMLSchema was to design, implement and run a suitable software system which was given the acronym ROAD – standing for Retrieval of Astronomical Data. Tests were to be done on the system and the outputs then analysed.

To enable this testing an XMLSchema was to be created and datasources made available.

Chapter 4

Software Development

4.1. Actual Data Sources

At this point in the research the data sources to be used to demonstrate the software needed to be determined to enable the construction of queries, the query results object and the transformation files (XSLT) to produce an instance of the XML Document containing the retrieved data. Also an astronomical target needed to be decided upon.

It proved to be very difficult to obtain access to any data from scientific bodies such as the ATC and it became necessary to resort to publically available data sources. The ones used are listed below and they are all sources accessible through the web via web services.

The three data sources chosen were picked both because they were large repositories of data and also because the queries returned clearly different content. They are described in Chapter 2 as well as in Appendix 18:

The star Tau Ceti (HIP 8102) was chosen as the target object. This was because it is relatively close and has been studied frequently with the result that there is a variety of data available.

For this proof of concept, using the target of Tau Ceti there were four entries into the query response object. This was the data used for the construction of the XMLDocument via the XSLT files and XMLQuery definition. There were four queries from the three datasources due to the fact that in addition to the XML returned from the OEC datasource it was also possible to return the HTML (containing useful diagrams) and use both in the outputs:

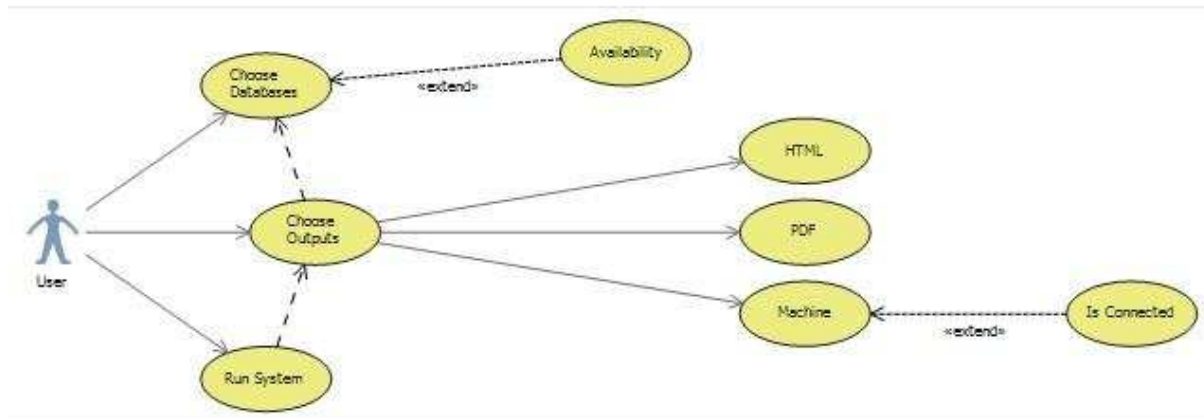
It is worth noting that all the data queries are optional and chosen by the user, therefore it was possible to use any number of the four data queries that were available. Also, the data returned from these databases were from observations taken in the visible and infra-red ranges, therefore these were the nodes that were developed in the XMLSchema used in this research.

4.2. System Implementation

The work involved in the design of the software used in this research was as follows.

A Use Case was drawn:

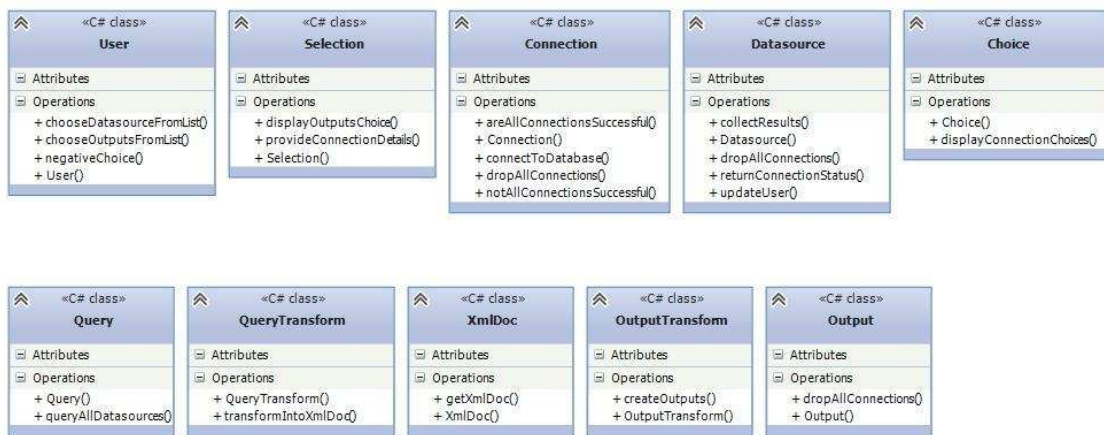
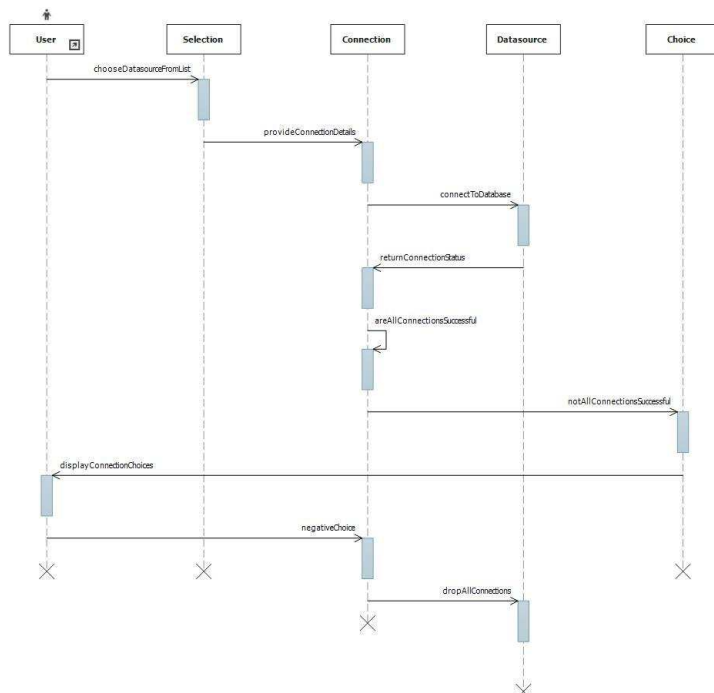
Figure D.3: ROAD System Use Case



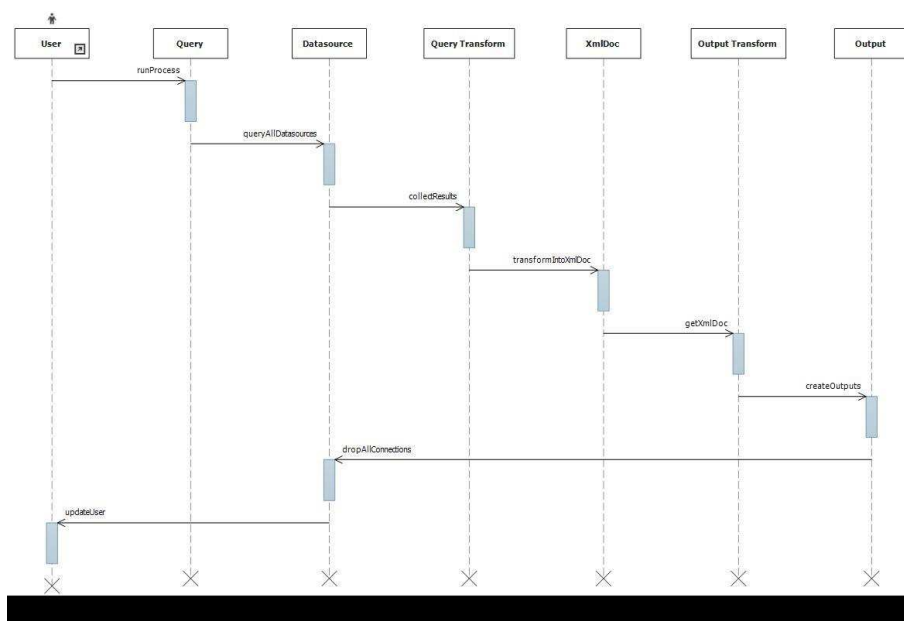
Then a textual statement of the software process was taken from the Software Requirements Specification (Section 3.3.2. of this document):

‘The user first chooses the data sources and then the desired outputs. An astronomical object is then selected. The database connections are tested. If not all are connected the user is given the choice to abort or to carry on with those successfully connected. The system queries the databases, transforms the returned results and puts them into a new XMLDocument. The XMLDocument is then used to produce the requested outputs. A log file is kept for all actions and errors.’

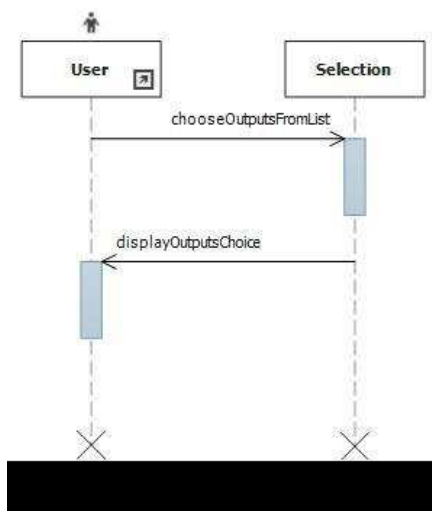
From this statement textual analysis was used to produce a UML Class Diagram and three UML Sequence Diagrams which show the sequence of operations for all three actions carried out by the ROAD application (to query the datasources chosen by the user, process the data retrieved from the datasources and to output the data in useable formats). These diagrams, which are shown below, formed the basis for implementation in that through them, classes are named and their methods defined. Final implementation never ends up the same as the diagrams, but then they are intended as a guide and are not supposed to be slavishly followed.

Figure D.4: ROAD - Other Initial UML Diagrams**ROAD Class Diagram****ROAD ChooseDataSources**

ROAD ProcessData



ROAD ChooseOutputs



4.2.1. Inputs

The inputs are pre-set in so far as the data source queries are hard coded into the application. This was seen to be acceptable for an initial proof of concept. It is however recognised that an application for general use would have a way of entering queries at run time, such as through the web.config file or through a text box within which a user could make a database connection and enter a full query string. It was seen to be essential that where multiple sets of the same kind of

data and object are retrieved, such as in this case, these are returned in the query result in a distinct way. Only when being prepared for output is the data finally merged. Therefore when querying, it was decided that each database must have its own area within the query report object by initially storing the data source responses as items in a Sorted List. The data in each item is wrapped in a simple XML format.

Log and Error output functionalities were also implemented. Additionally, it was decided not to give the user the choice to abort or to carry on in case of a data source connection failure as any error would be output to the log file which would give sufficient information for the proof of concept software runs.

The user web screen has a simple, no frills design but gave all the functionality that was required. It ensured that the user had the choice of up to four data queries and up to three output type choices. The choices of both queries and outputs are chosen from their respective lists and subsequently confirmed. The option to remove choices is also provided. Once the choices are made and confirmed the retrieval button can be used, which begins the software run that both produces the XMLDocument and the output files.

There was no choice of the target, it being set as the stellar system of Tau Ceti by default. The reason for this lack of target choice was due to time constraints and the fact that a variety of targets was not a high aspiration of the demonstration of this research project, the main aims being those of designing an XMLSchema and demonstrating the transformations through the use of lists and XSLT files. Again it was recognised that in any general use this software would need to enable the user to enter target detail in the form of name, position or catalogue ID.

4.2.2. Transformation

The whole purpose of the transformation stage is to produce an XMLDocument that is validated by the XMLSchema used in this research which is a file called 'astronomy.xsd'.

The XMLSchema was developed with the detail of the Observational Astronomy node being built alongside the XSLT files to be used to help transform the query results into a format that can be assembled into the XMLDocument. This provides for the production of the particular instance of

the XMLDocument holding the data about the target object Tau Ceti. There was one XSLT file required for each of the three data sources which were the subject of the four data source queries. This enables the correct transformation of all the data held in the query result object to enable validation via the XMLSchema.

A simplified diagram of the structure is shown in Appendix 1, whereas the detail of the astronomy.xsd XMLSchema code is shown in Appendix 2.

The results from each query are stored within the XMLDocument as individual nodes within an Observational Astronomy node. This enables distinction of data of the same astronomical object from multiple databases.

The XMLSchema was developed far enough to accommodate the data for this proof of concept. Therefore in the schema the node of Observational Astronomy was further developed along with its child nodes of 'Visible' and 'Infra-Red' as these were the relevant nodes to hold the test data.

Once the XMLDocument had been produced and it had been successfully validated by the XMLSchema the outputs could then be produced according to the choices of the user.

4.2.3. Outputs

The software that produces the PDF and HTML output files was written in a way to adapt the output page layouts according to the number of queries chosen by the user. The purpose of the data outputs in this proof of concept are just to show the kind of outputs that are possible, they are not supposed to attempt a comprehensive output structure for all possible queries.

The production of the output files adapts the page layouts of each output file according to the number of data queries returned, whether it be just one query that is chosen by the user or any number up to the maximum of four that can be chosen in this ROAD software.

The Machine output choice could also have been transformed into any format that would be readable dependent upon a given target machine. This could be in any text format, CSV for example. It may also be in XML format. This format would evidently be extremely variable

requiring a separate XSLT file for each target machine to generate the individual data format that each machine would require. Therefore it was decided, as the output was already in XML format for the purposes of these test runs the Machine output would just be the XMLDocument saved to file without any transformations.

As above, in general use the application would be enhanced so that it would be possible to specify XSLT files via the web.config or through a text box within which a user could specify the location of an XSLT file, for the purposes of creating their own personalised outputs.

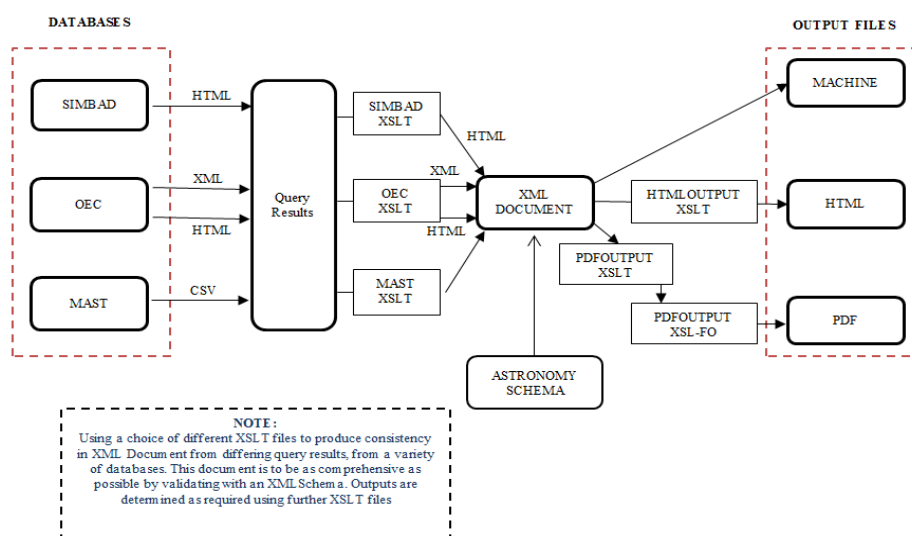
Whilst the XMLSchema 'astronomy.xsd' is one of the outputs of the research, for the purposes of the user 'outputs' in this context means the chosen outputs of the user which comprises a choice of one or all of PDF, HTML and XML.

It is clearly the case that the PDF and HTML outputs are for human use and the Machine (XML) output mainly for machine to machine transfer.

4.2.4. Completed ROAD system

On completion of the system implementation the final system structure that was actually created, shown in the diagram below, was not too different from the original system design.

Figure D.5: Final ROAD System Design



The final developed code resulted in a simpler design than originally planned, utilizing just five classes. The careful planning which initially suggested more classes gave a clear initial structure and was easily refined down to these five during the implementation process. One of these classes was the user interface ‘road.aspx.cs’ (See Appendix 9). This class enabled user interaction to choose options and carry out the system runs. It also produced the log file stream and carried out the XMLSchema validation checks. The road.aspx.designer.cs companion class produced the visual user interface. Table 8 below shows all the class files developed for the ROAD application.

Table 8: Actual ROAD Classes

File name	Functionality
road.aspx.cs	Enables user interaction, creates log file and utilises schema (See Appendix 9)
road.aspx.designer.cs	Produces user interface
Query.cs	Queries databases and puts all results into a SortedList (See Appendix 10)
QueryTransform.cs	Carries out data transform from SortedList into an XmlDocument (See Appendix 11)
OutputTransform.cs	For each selected output choice, does transforms and creates output file (See Appendix 12)

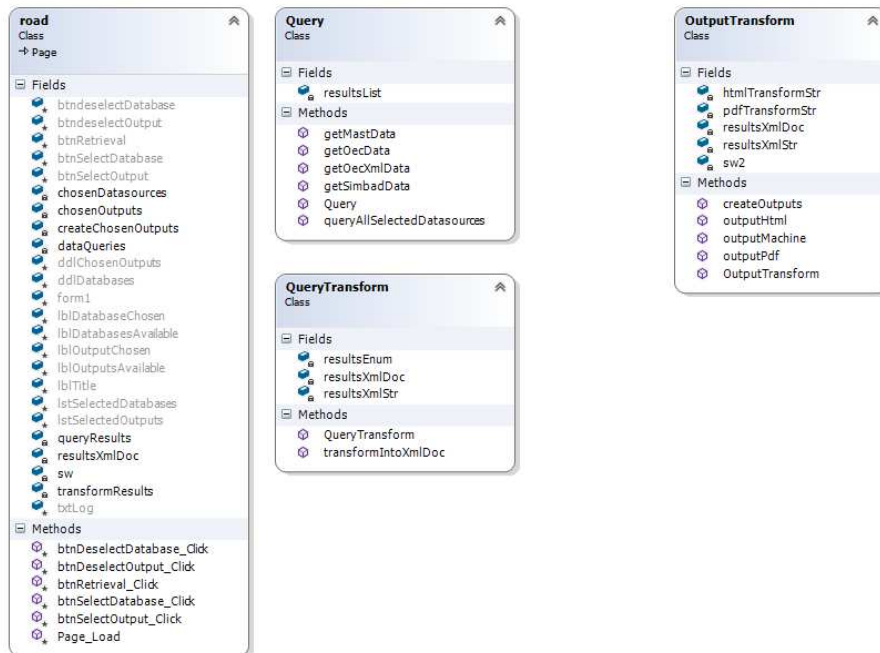
Other files created for the system are listed in Table 9 below:

Table 9: Other ROAD Files

File name	Functionality
astronomy.xsd	The XMLSchema (See Appendix 2)
mastQuery.xslt	Input transformation file (See Appendix 13)
oecQuery.xslt	Input transformation file (See Appendix 14)
simbadQuery.xslt	Input transformation file (See Appendix 15)
htmlOutput.xslt	Output transformation file (See Appendix 16)
pdfOutput.xslt	Output transformation file (See Appendix 17)

Figure D.6 below is a UML diagram of the final class structures that were developed and implemented:

Figure D.6: ROAD UML Class Diagram



Chapter 5

Software System (ROAD) Runs

The data returned by the ROAD database queries was to be structured in a standard way by the ROAD application in accordance with XMLSchema validation. The data from each query result was to be stored in the results XMLDocument in a way that separates it from all the other query results but in a standard structure which is the same as the others. For each database query executed during the ROAD application run there was to be an area created in the XMLDocument within which to put the data that was retrieved. This was to make it easy to extract the data later (in this case later in the ROAD application run) from the XMLDocument when it was desired to create an output document for human or machine use.

The files created as output (html, pdf and others) by the ROAD application were to be simple examples of how the data in the different areas could be combined together into a single output document. This was intended to show that data from a single XMLDocument can be transformed into a variety of output document types using some or all (depending on ROAD application user choice) of the data held in the document.

The XMLDocument structure was designed to enable the holding of metadata for easy retrieval and also large blocks of data in string or BLOB format. This means that all types of data could, if desired be retrieved into the XMLDocument such as URLs, images, spreadsheets and other datasets.

Given that the available database data for these queries available to be used was either of visual or infra-red data from astronomical observations, the areas of the XMLSchema required were within the visual and infra-red child nodes of 'Observational Data' which consequently were the areas fully developed in the astronomy.xsd schema. The potential for the creation of other nodes would still remain available. The structure was meant to provide maximum flexibility of types of data that are retrievable along with a rigid structure for metadata storage which was to be saved along with each database query result.

5.1. The purpose and planning of the System runs

The system runs were to be tests to demonstrate the correct functioning of the developed software system. Referring to the Software Requirements Specification in Section 3.3.2 of this document, the test strategy was to achieve a number of objectives relating to these requirements, as listed in Table 10 below:

Table 10: Objectives of ROAD system runs

Objective	Requirements Specification	Test Outcome
Carry out queries of multiple databases	Process data Use Case	Production of XMLDocument containing multiple database data
Combine multiple database queries into a single query result list	Process data Use Case	Production of XMLDocument containing multiple database data
Create an XMLDocument from the results data retrieved and save to file	Process data Use Case	Production of XMLDocument containing multiple database data
Validate that XMLDocument against the XMLSchema	Process data Use Case	Production of output files of chosen output types
Produce output files, subject to successful XMLDocument validation, that utilizes the XMLDocument data according to the needs of the user	Process data Use Case	Production of all the output file types chosen by the User
Prevent output file production if the XMLDocument is invalid and inform the user of the precise validation fault of the XMLDocument	Process data Use Case	Production of XMLDocument containing multiple database data but no chosen output files produced. Log file contains error that describes the reason for the XMLDocument being invalid
Enable the user to choose the databases and outputs from lists	Choose data sources Use Case Choose outputs Use Case	Successful production of the validated XMLDocument and chosen output files

In this way the creation of the XMLDocument containing data from multiple data sources subject to validation by correct operation of the extensible XMLSchema astronomy.xsd, could be tested as well as the creation output documents (when the XMLDocument successfully validated!), each document containing data taken from all of the chosen datasources.

The test parameters were for the user to select varying multiple databases and also varied combinations of output types. A limited list of accessible databases and output types was to be provided as part of the Retrieval of Astronomical Data (ROAD) application. In addition to this the ROAD system would be configured so that one test would fail due to an invalid XMLDocument being created, this to test and demonstrate the control of the XMLSchema over

output file production. At the end of successful testing there were to be a number of log files and XMLDocuments produced along with output files produced from validated XMLDocuments, according to the choices of the User.

It was decided that there would be five runs of the ROAD application. This would be enough runs to show different data source combinations and XMLSchema control. Four would show the system handling various choices of inputs and outputs. The input data types being retrievable by choices from the input list were those of XML, HTML and CSV. The output types available were those of PDF, HTML and Machine (basically XML output).

The final run was designed to force a schema validation error to show its effectiveness in the prevention of the creation of malformed data outputs.

These five test runs were considered sufficient to show the ability of the ROAD system to produce varied output from some or all of the available databases whilst also demonstrating the control of data output structure through the use of the XMLSchema.

It should be noted that the output XSL-FO files were rendered into PDF using a free on-line service by Attic. (XSL-FO Rendering Online, 2015)

The five runs chosen were as follows:

Table.11: Table of ROAD Runs

Number of Run	Choice of Sources	Choice of Outputs
1	MAST, SIMBAD, OEC_XML, OEC	HTML, PDF, MACHINE
2	MAST, SIMBAD, OEC_XML	HTML
3	SIMBAD, OEC_XML, OEC	PDF
4	MAST, OEC_XML	MACHINE
5	MAST, SIMBAD, OEC_XML, OEC. (Set to produce schema error)	HTML, PDF, MACHINE

To create an expected error in run five, a change was made to the mastQuery.xslt file which normally organizes the query results from MAST into a correct format for inclusion into the XMLDocument. The correct structure defined by the XMLSchema is for all the MAST returned

data to be within the following node path:

astronomy/observationalastronomy/infrared/iobservationdata

The XSLT file was changed so the returned data was returned within a structure that reflected the following node path:

astronomy/observationalastronomy/infrared/visualobservationdata

This is invalid according to the XMLSchema.

The system log and XMLDocument files produced were saved to the local machine at the location of 'C:/roadLogs/' and the output files were saved to 'C:/roadLogs/outputs/'.

The runs were then carried out and the outputs stored. The table in Appendix 21 summarizes the actions of the system runs.

5.2. Results of the System runs

Run1

This run proceeded and completed successfully. All four data queries were chosen and all three outputs also. All three output files were produced along with the XMLDocument file and the log file. The PDF file was rendered and saved using the Attic online renderer. (XSL-FO Rendering Online, 2015)

The total time of the system run was approximately two seconds.

Run2

This run proceeded and completed successfully. The data queries of MAST, SIMBAD and OEC_XML were chosen and the output of HTML. The HTML output file was produced along with the XMLDocument file and the log file.

The total time of the system run was approximately two seconds.

Run3

This run proceeded and completed successfully. The data queries of SIMBAD, OEC_XML and OEC were chosen and the output of PDF. The PDF output file was produced along with the XMLDocument file and the log file. The PDF file was rendered and saved using the Attic online renderer. (XSL-FO Rendering Online, 2015)

The total time of the system run was approximately one second.

Run4

This run proceeded and completed successfully. The data queries of MAST and OEC_XML were chosen and the output of MACHINE. The MACHINE output file was produced along with the XMLDocument file and the log file.

The total time of the system run was approximately one second.

Run5

This run proceeded and completed successfully. All four data queries were chosen and all three outputs also. The XMLDocument file and the log file were successfully produced and saved. No output files were produced.

The total time of the system run was approximately two seconds.

The results of the ROAD system runs are summarised in Table 12 below, the analysis of these results is to be found in the next chapter, Chapter 6:

Table.12: Table of ROAD Run Results

Number of Run	Choice of Sources	Choice of Outputs	Actual Outputs	Log Outputs	Date-Time
1	MAST, SIMBAD, OEC_XML, OEC	HTML, PDF, MACHINE	resultsXmlDocument_15 03 2015 19-08-48.xml htmlOutput_15 03 2015 19-08-50.html machineOutput_15 03 2015 19-08-50.xml pdfOutput_15 03 2015 19-08-50.fo run1output.pdf (using Attic)	myTest_15 03 2015 19-08-48.txt	15 03 2015 19-08-48
2	MAST, SIMBAD, OEC_XML	HTML	resultsXmlDocument_15 03 2015 19-13-28.xml htmlOutput_15 03 2015 19-13-30.html	myTest_15 03 2015 19-13-28.txt	15 03 2015 19-13-28
3	SIMBAD, OEC_XML, OEC	PDF	resultsXmlDocument_15 03 2015 19-17-05.xml pdfOutput_15 03 2015 19-17-06.fo run3output.pdf (using Attic)	myTest_15 03 2015 19-17-05.txt	15 03 2015 19-17-05
4	MAST, OEC_XML	MACHINE	resultsXmlDocument_15 03 2015 19-24-50.xml machineOutput_15 03 2015 19-24-51.xml	myTest_15 03 2015 19-24-50.txt	15 03 2015 19-24-50
5	MAST, SIMBAD, OEC_XML, OEC. (Set to produce schema error)	HTML, PDF, MACHINE	resultsXmlDocument_15 03 2015 19-35-19.xml	myTest_15 03 2015 19-35-19.txt	15 03 2015 19-35-19

The screenshot and log file output (myTest file) for each run are to be found in Appendix 3.

Chapter 6

Data Analysis

Dependent on the choices of the user on each run, there were up to four separate datasource query results returned and held in the ROAD system application Sorted List object ‘queryResults’, which is declared in the class ‘road.aspx.cs’. The data is too large to be included here or in the appendices, but all four can be viewed in digital format as text, in the file ‘queryResult.txt’ in the flash drive or cd medium provided along with this dissertation document. From this collection of data some or all, as required, was passed into the XMLDocument.

The results were looked at in more detail and the findings produced. It needs to be mentioned at this point that all the files produced, along with the system application source files, are available for viewing in digital format in either flash drive or cd medium provided along with this dissertation document.

6.1. Detailed Analysis of the results

The analysis of each set of results, the XMLDocument produced and subsequent output files, are set out below.

Run1

The validated XMLDocument ‘resultsXmlDocument_15 03 2015 19-08-48.xml’ had the following structure:

```
<?xml version="1.0" encoding="UTF-8"?>
- <astronomy xmlns:noNamespaceSchemaLocation="file:///C:/ROAD//RetrievalOfAstronomicalData//RetrievalOfAstronomicalData//astronomy.xsd">
  <astrophysics/>
  <astrobiology/>
  <astrochemistry/>
  <archeoastronomy/>
  <planetaryscience/>
  + <observationalastronomy>
  + <observationalastronomy>
  + <observationalastronomy>
  + <observationalastronomy>
</astronomy>
```

The four queries used resulted in four ‘observationalastronomy’ nodes being created. Expanding the document structure further it can be seen that there are three visible types and one infrared type of observation.

```

<?xml version="1.0" encoding="UTF-8"?>
<astronomy xmlns:noNamespaceSchemaLocation="file:///C:///ROAD//RetrievalOfAstronomicalData//RetrievalOfAstronomicalData//astronomy.xsd">
  <astrophysics/>
  <astrobiology/>
  <astrochemistry/>
  <archeoastronomy/>
  <planetaryscience/>
  - <observationalastronomy>
    + <visible>
  </observationalastronomy>
  - <observationalastronomy>
    + <visible>
  </observationalastronomy>
  - <observationalastronomy>
    + <visible>
  </observationalastronomy>
  - <observationalastronomy>
    + <infrared>
  </observationalastronomy>
</astronomy>

```

Drilling further down by expanding one of the visible nodes and the infrared node the various nodes that hold data can be seen.

```

<?xml version="1.0" encoding="UTF-8"?>
- <astronomy xmlns:noNamespaceSchemaLocation="file:///C:///ROAD//RetrievalOfAstronomicalData//RetrievalOfAstronomicalData//astronomy.xsd">
  <astrophysics/>
  <astrobiology/>
  <astrochemistry/>
  <archeoastronomy/>
  <planetaryscience/>
  - <observationalastronomy>
    - <visible>
      + <device>
      + <location>
      + <target>
      + <targetephemeris>
      + <datetime>
      + <weather>
      + <observer>
      + <visibleobservationdata>
      + <images>
      + <comments>
    </visible>
  </observationalastronomy>
  - <observationalastronomy>
    + <visible>
  </observationalastronomy>
  - <observationalastronomy>
    + <visible>
  </observationalastronomy>
  - <observationalastronomy>
    - <infrared>
      + <device>
      + <location>
      + <target>
      + <targetephemeris>
      + <datetime>
      + <weather>
      + <observer>
      + <irobservationdata>
      + <images>
      + <comments>
    </infrared>
  </observationalastronomy>
</astronomy>

```

Looking more closely at the infrared node, for example

```
<infrared>
<device>
<manufacturer>
<model>=HAST Repository</model>
</device>
<location>
<country>=USA</country>
</location>
<target>
<targetname>=Tau Ceti</targetname>
<targetposition>
</targetposition>
<targetphenomenis>
<phenomenis>=</phenomenis>
</targetphenomenis>
<datetime>
<startdate>=15 03 2015</startdate>
</datetime>
<weather>
<weatherdesc>=</weatherdesc>
</weather>
<observer>
<formname>
<username>=</username>
<contact>
</observer>
<observationaldata>
<thetaima>
<thetaeastunit>
<thetaeastlength>
<description>=CSV</description>
<thetaunit>
<elementary>
string,string,rad,deg,integer,datetime,datetime,float,string,string,float,integer,datetime,string,integer,float[U33,U70IM,C1]-U8I800060-1543,00 08 35,281,-15 27 20,22,35,1997-05-07 08:00:54,400,00,WPCF2_WF3,81W4W,8012,606,6346,1998-05-07 13:06:16,U33L,CZDPA,15,353 NSZXARKQJ,ANY,00 02 12,868,-15 32 28,25,2,1998-09-26 01:54:44,1998-09-26 02:11:48,1023.950,NICMOS,NIC2,F22PM,22181,699,8082,1998-09-26 18:27:26,NSZXARKQJ,51,249 NAFXGFSQJ,ANY,00 02 12,868,-15 32 28,25,1,1998-09-26 06:36:14,1998-09-26 06:39:42,447,576,NICMOS,NIC2,F22PM,22181,699,8063,1998-09-26 17:32:24,NAXFGFSQJ,ANY,00 02 12,868,-15 32 28,25,1,1998-09-26 06:50:07,447,576,NICMOS,NIC2,F22PM,22181,699,8063,1998-09-26 18:06:37,NAXFGFSQJ,51,249 NSXZALXSQJ,ANY,00 02 12,868,-15 32 28,25,2,1998-09-26 03:26:25,1998-09-26 17:29,1023.950,NICMOS,NIC2,F22PM,22181,699,8082,1998-09-26 18:42:19,NSZXASQJ,51,249 NSZXANSQJ,ANY,00 02 12,868,-15 32 28,25,1,1998-09-26 05:07:11,1998-09-26 05:24:17,1023.950,NICMOS,NIC2,F22PM,22181,699,8082,1998-09-26 18:49:40,NSZXANSQJ,51,249 NSZXARQJ,ANY,00 02 12,780,-15 32 29,77,2,1998-09-26 01:34:28,1998-09-26 01:51:32,1023.950,NICMOS,NIC2,F22PM,22181,699,8082,1998-09-26 15:54:00,NSZXARKQJ,51,249 NSXZALXSQJ,ANY,00 02 12,780,-15 32 29,77,2,1998-09-26 03:10:09,1998-09-26 13:10,1023.950,NICMOS,NIC2,F22PM,22181,699,8082,1998-09-26 16:31:04,NSXZALXSQJ,51,249 NSZXANSQJ,ANY,00 02 12,780,-15 32 29,77,2,1998-09-26 04:46:57,1998-09-26 05:04:01,1023.950,NICMOS,NIC2,F22PM,22181,699,8082,1998-09-26 17:09:52,NSZXANSQJ,51,249
</observationaldata>
<images>
<files>=</files>
<filename>
<imagename>
<commented>
</commented>
</images>
<comments>=For further info check the data</comments>
</comments>
</infrared>
```

Data can be held in any of these nodes, which is dependent on the XSLT file used. In this run, as in the others some of these nodes were used, the main bulk of the data received being placed into the ‘visibleobservationdata/thedatavisible’ and, as above, the ‘irobservationdata/thedatair’ nodes. The fully expanded xml showing all the data can be seen in Appendix 4. In places HTML Special Entities have been used to temporarily replace ‘<’ and ‘>’ tags in areas where they would otherwise cause confusion to the xml structure. In addition the main blocks of data in the SIMBAD and OEC query responses have been ‘CDATA’ wrapped to prevent any problems with any other possible XML invalid characters.

This XMLDocument had been constructed by first putting the four returned database queries as four separate objects into a single list and then using the XSLT files to combine them into that XMLDocument. The document was output as a file and then validated against the XMLSchema ‘astronomy.xsd’. The validation code can be found in the source code at line 142 in the class file ‘road.aspx.cs’.

The outputs of Machine, PDF and HTML were all successfully produced.

Machine output for the purposes of this research was just the XMLDocument passed straight through without change. It could easily be changed to another XML format in future circumstances to suit any receiving machine but for these purposes, as it was already XML format, no change was made and it was passed straight through to create the output file

‘machineOutput_15 03 2015 19-08-50.xml’.

HTML output created the file ‘htmlOutput_15 03 2015 19-08-50.html’ from the XMLDocument via use of the ‘htmlOutput.xslt’ file. This web page contains selected information from all four datasource queries. See the image of the webpage in Appendix 5. In this web page there are four sections one below the other consisting of HTML data from SIMBAD, XML data from OEC_XML, HTML data from OEC and CSV data from MAST in that order, all transformed into HTML. Note that the data has been filtered in each case to display only information that is wanted. The filtering and transform being carried out both by the htmlOutput.xslt file and by the system application class ‘OutputTransform.cs’. If you open the actual file from the digital resource you will be able not only to see the page better (this applies to all the files), but you will also be able to use the two links that are on the page. Note that the HTML output files are best viewed in Firefox.

PDF output created the ‘pdfOutput_15 03 2015 19-08-50.fo’ file. This was then separately rendered using the Attic online renderer (XSL-FO Rendering Online, 2015) to produce the ‘run1output.pdf’ file which can be seen in Appendix 6. This carries the same information as the previous HTML page except, obviously, in PDF format. As in the HTML output above, opening the actual PDF file will give a better view of the PDF pages and also enable the link to be used.

All of this activity was saved to the system application log file ‘myTest_15 03 2015 19-08-48.txt’. The content of this log file can be seen in Appendix 3. Run1.

A lot of that written above regarding Run1 also applies to the other runs so, in the interests of concision, the remaining descriptions just contain the differences.

Run2

Three queries were asked for in this run, two visual and one of infrared data and therefore three nodes consisting of these types in the XMLDocument ‘resultsXmlDocument_15 03 2015 19-13-28.xml’ were expected, as is shown to be the case below:

```
<?xml version="1.0" encoding="UTF-8"?>
- <astronomy xmlns:noNamespaceSchemaLocation="file:///C:/ROAD//RetrievalOfAstronomicalData//RetrievalOfAstronomicalData//astronomy.xsd">
  <astrophysics/>
  <astrobiology/>
  <astrochemistry/>
  <archeoastronomy/>
  <planetaryscience/>
  - <observationalastronomy>
    + <visible>
  </observationalastronomy>
  - <observationalastronomy>
    + <visible>
  </observationalastronomy>
  - <observationalastronomy>
    + <infrared>
  </observationalastronomy>
</astronomy>
```

A single output type of HTML was called for and that was produced in file 'htmlOutput_15 03 2015 19-13-30.html'.

This web page contains selected information from the three datasource queries. See the image of the webpage in Appendix 7. In this web page there are three sections one below the other consisting of HTML data from SIMBAD, XML data from OEC_XML and CSV data from MAST in that order, all transformed into HTML

This activity was saved to the system application log file 'myTest_15 03 2015 19-13-28.txt'. The content of this log file can be seen in Appendix 3. Run2.

Run3

Three queries were asked for in this run, all being of visual data and therefore three nodes consisting of visual type in the XMLDocument 'resultsXmlDocument_15 03 2015 19-17-05.xml' were expected, as is shown to be the case below:

```
<?xml version="1.0" encoding="UTF-8"?>
- <astronomy xmlns:noNamespaceSchemaLocation="file:///C:/ROAD//RetrievalOfAstronomicalData//RetrievalOfAstronomicalData//astronomy.xsd">
  <astrophysics/>
  <astrobiology/>
  <astrochemistry/>
  <archeoastronomy/>
  <planetaryscience/>
  - <observationalastronomy>
    + <visible>
  </observationalastronomy>
  - <observationalastronomy>
    + <visible>
  </observationalastronomy>
  - <observationalastronomy>
    + <visible>
  </observationalastronomy>
</astronomy>
```

A single output type of PDF was called for and that was produced in file 'pdfOutput_15 03 2015 19-17-06.fo'. This was then separately rendered using the Attic online renderer (XSL-FO Rendering Online, 2015) to produce the 'run3output.pdf' file which can be seen in Appendix 8.

This activity was saved to the system application log file 'myTest_15 03 2015 19-17-05.txt'. The

content of this log file can be seen in Appendix 3. Run3.

Run4

Two queries were asked for in this run, one visual and one of infrared data and therefore two nodes consisting of one visual type and one of infrared type in the XMLDocument 'resultsXmlDocument_15 03 2015 19-24-50.xml' were expected, as is shown to be the case below:

```
<?xml version="1.0" encoding="UTF-8"?>
- <astronomy xmlns:noNamespaceSchemaLocation="file:///C:/ROAD//RetrievalOfAstronomicalData//RetrievalOfAstronomicalData//astronomy.xsd">
  <astrophysics/>
  <astrobiology/>
  <astrochemistry/>
  <archeoastronomy/>
  <planetaryscience/>
  - <observationalastronomy>
    + <visible>
  </observationalastronomy>
  - <observationalastronomy>
    + <infrared>
  </observationalastronomy>
</astronomy>
```

A single output type of MACHINE was called for and that was produced in file 'machineOutput_15 03 2015 19-24-51.xml'.

This activity was saved to the system application log file 'myTest_15 03 2015 19-24-50.txt'. The content of this log file can be seen in Appendix 3. Run4.

Run5

All four queries were asked for in this run, three visual and one of infrared data and therefore four nodes consisting of three visual types and one of infrared type in the XMLDocument 'resultsXmlDocument_15 03 2015 19-35-19.xml' were expected, as is shown to be the case below:

```
<?xml version="1.0" encoding="UTF-8"?>
- <astronomy xmlns:noNamespaceSchemaLocation="file:///C:/ROAD//RetrievalOfAstronomicalData//RetrievalOfAstronomicalData//astronomy.xsd">
  <astrophysics/>
  <astrobiology/>
  <astrochemistry/>
  <archeoastronomy/>
  <planetaryscience/>
  - <observationalastronomy>
    + <visible>
  </observationalastronomy>
  - <observationalastronomy>
    + <visible>
  </observationalastronomy>
  - <observationalastronomy>
    + <visible>
  </observationalastronomy>
  - <observationalastronomy>
    + <infrared>
  </observationalastronomy>
</astronomy>
```

All three output types of HTML, PDF and MACHINE were called but no output files were produced. This is because an xml schema validation error had intentionally been caused to test the validation process was operating. The mastQuery.xslt file had been altered so that instead of creating the correct structure in the infrared node thus:


```

- <observationalastronomy>
  - <infrared>
    + <device>
    + <location>
    + <target>
    + <targetephemeris>
    + <datetime>
    + <weather>
    + <observer>
    + <irobervationdata>
    + <images>
    + <comments>
  </infrared>
</observationalastronomy>

```

An invalid structure was created, like so:

```

- <observationalastronomy>
  - <infrared>
    + <device>
    + <location>
    + <target>
    + <targetephemeris>
    + <datetime>
    + <weather>
    + <observer>
    + <visualobservationdata>
    + <images>
    + <comments>
  </infrared>
</observationalastronomy>
</astronomy>

```

This structure was known by the system to be invalid when it was tested against the XMLSchema ‘astronomy.xsd’ causing an error to be thrown. Consequently no outputs were allowed to be created.

The actual automatic error produced was output to the log file. The error is both precise and informative. The content of the log file is shown here also, thus:

```

=====
ROADS beginning data retrieval 15/03/2015 19:35:19
=====
Querying for data...

Querying for Mast
Querying for OEC
Querying for OEC_XML
Querying for Simbad

Query results returned
The element 'infrared' has invalid child element 'visualobservationdata'. List of possible elements expected: 'irobervationdata'.
resultsXmlDoc did not validate

XMLDocument created: C://roadLogs//resultsXmlDocument_15 03 2015 19-35-19.xml

*** Error *** No file outputs due to XMLDocument schema validation error
=====
ROADS finished data retrieval 15/03/2015 19:35:21
=====

```

This activity was saved to the system application log file ‘myTest_15 03 2015 19-35-19.txt’. The content of this log file can also be seen in Appendix 3. Run5.

Chapter 7

Discussion

7.1. Previous research

When setting out on this research the review of available literature demonstrated both the huge amount of astronomical data available and the rapid rate of its increase. Additionally, the ability to store data digitally had increased dramatically in the last 30 years (Lopez, 2014). Given the fact that nearly all astronomical data is now held digitally, the retrieval of data appeared to be a more easily accessible process. However there was also evident a lack of any widespread standard of structured data recording. Given this situation, a way of retrieving the very disparate astronomical data through the conduit of a widespread standard data structure would greatly benefit users of the data. This research has been carried out with the intention of working towards such a solution.

The literature review showed that the only XMLSchemas created so far have been narrow in their design, having a root node of just one area of astronomy, such as the Openastronomylog schema which has effectively a root node of visual observational astronomy (Openastronomylog, 2014). This schema is basically a root node called ‘Base’ under which there are a few child nodes, all to do with observational astronomy in the visual spectrum.

This is an open source schema which is the best of that kind available and these child nodes have been well developed in depth but there is a problem which is that this is a structure based on data about observable astronomical targets. It cannot be expanded to include any other areas of astronomy.

As has been seen in Table 5 in the review earlier on (section 2.3.8 of this document), there were no schemas found that did extend or were extensible across the whole of ‘Observational Astronomy’ data, let alone also the other five areas that make up between them the whole range of astronomical data. It was concluded that the only way to achieve this was to design a new schema containing a structure capable of such expansion.

As will now be discussed, this research has provided a better schema in terms of its extensibility and potential range across the breadth of astronomical data than any other that were found during the review that was carried out (Chapter 2 of this document).

7.2. Summary of the main experimental results

An application was written and implemented to enable a user to choose multiple databases to query for astronomical data and combine this data into an XMLDocument of a defined structure which could be validated against a custom schema. If the validation was successful the XMLDocument was then converted into output file types that the user had also chosen from a list.

An XMLSchema ‘astronomy.xsd’ was created to enable validation of the XMLDocuments created.

Five test runs were carried out. All the runs correctly produced one XMLDocument each containing data from the datasource queries that were chosen by the user. Output files were also produced from each run as follows:

Table 13: Table of Main Results

Test Run	User choice of datasources	User choice of Output types	Actual File Output types
1	MAST, SIMBAD, OEC_XML, OEC	HTML, PDF, MACHINE	HTML, PDF, MACHINE
2	MAST, SIMBAD, OEC_XML	HTML	HTML
3	SIMBAD, OEC_XML, OEC	PDF	PDF
4	MAST, OEC_XML	MACHINE	XML
5	MAST, SIMBAD, OEC_XML, OEC. (Set to produce schema error)	HTML, PDF, MACHINE	NO FILES CREATED

Therefore the first four test runs correctly produced the output files requested by the user. The final fifth run did produce a schema error as expected and correctly, in accordance with expectations, did not create any user output files.

7.3. Interpretation of the results

An XMLDocument containing the data from the datasources chosen by the user was successfully created and saved to file during each test run. This showed that the datasource queries returned results and then the transform of them into an XMLDocument worked well and to expectations.

The first four test runs correctly created and saved to file the output files requested by the user. This showed that the transform from the XMLDocument to the output files worked well and to expectations.

The final fifth run produced a schema error as expected and therefore quite correctly did not create any user output files. An informative error message describing the reason for the schema validation failure was output to the log file. This showed that the XMLSchema that has been created was working properly to enable successful validation.

The application created to carry out the test runs performed well and to expectations, the run time being under two seconds.

7.4. The significance of the results

There were two main objectives of this research:

- To show that from a variety of astronomical data sources and by using XML Technologies an advanced construct of usefully structured data could be produced in an XML Document.
- To develop an XMLSchema in such a way that it not only handles the variety of data used in this research but is capable of being extended to include the full breadth of astronomical data storage content in the future.

The results of the research were to be shown both in the XMLDocuments generated from the spread of datasource query data and also by the extent to which the XMLSchema file ‘astronomy.xsd’ had been developed within a structure that is of an extendable design. Also, to show that the XMLDocuments produced are useful, output files were produced from them in HTML, XML and PDF formats.

The use of XML Technologies was demonstrated to be a valid way to convert retrieved astronomical data into an XMLDocument that was subject to validation, to ensure that the document conformed to a pre-defined structure. This is shown by the XMLDocument output files produced. This can be seen in Appendix 4 and by opening in the digital resource the ‘resultsXmlDocument’ xml files.

The XMLSchema structure was developed in the ‘astronomy.xsd’ file in sufficient detail to be able to validate the XMLDocuments produced during the five system runs whilst remaining extensible for further

areas of astronomy. This achievement is now discussed in more detail.

7.4.1. How the data in the XMLDocument is held, why it is useful, how the structure helps

For the purposes of this research a system application was developed for the ‘Retrieval of Astronomical Data (ROAD)’. To use this system to search for information the user had to choose any number of up to four data queries about a particular astronomical object – the star Tau Ceti. The queries returned, in string format, a variety of data types, specifically HTML, XML and CSV format. The task of the application was to combine the information returned by the selected datasource queries into a single XMLDocument that was built with a structure that was validated by an XMLSchema. This schema was also developed during this research.

The data received as query responses were in the form of blocks of HTML, XML or CSV information and were collected into a list with each query response held as a separate item. This ensured that each query response could subsequently be taken from the list, transformed in turn and added to the XMLDocument. The means of carrying out this transformation was through the use of XSLT files.

The subject matter of the data was either of visual observations or infrared observations, so those two areas of the schema were involved. They had a structure within the schema as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
- <astronomy xmlns:noNamespaceSchemaLocation="file:///C:/ROAD//RetrievalOfAstronomicalData//RetrievalOfAstronomicalData//astronomy.xsd">
  <astrophysics/>
  <astrobiology/>
  <astrochemistry/>
  <archeoastronomy/>
  <planetaryscience/>
  <observationalastronomy/>
</astronomy>

- <observationalastronomy>
  - <visible>
    + <device>
    + <location>
    + <target>
    + <targetephemeris>
    + <datetime>
    + <weather>
    + <observer>
    - <visibleobservationdata>
      <iscolour/>
      <magnitude/>
      <description/>
      <fileurlvisible/>
      <filenamevisible/>
      <thedatavisible/>
      <thedatavisibleblob/>
    </visibleobservationdata>
    + <images>
    + <comments>
  </visible>
</observationalastronomy>

- <observationalastronomy>
  - <infrared>
    + <device>
    + <location>
    + <target>
    + <targetephemeris>
    + <datetime>
    + <weather>
    + <observer>
    - <irobservationdata>
      <thetime/>
      <theintensity/>
      <thewavelength/>
      <description/>
      <fileurlir/>
      <filenameeir/>
      <thedatair/>
      <thedatairblob/>
    </irobservationdata>
    + <images>
    + <comments>
  </infrared>
</observationalastronomy>
```

Within either the <visible> or <infrared> node there are nodes for the storage of information about the observation including image references and then within the area of the green line there are nodes for the storage of actual observation data. These nodes for the actual results data as opposed to metadata about the observation are shown in Table 14 below:

Table 14: Table of astronomy.xsd node use

Node	Data held
'fileurlvisible' or 'fileurlir'	url link to the data
'filenamevisible' or 'filenameir'	reference to an actual file by name
'thedatavisible' or 'thedatair'	for the holding of string data
'thedatavisibleblob' or 'thedatairblob'	For BLOB data

In this research all the blocks of data were obtained as strings and so were put into either 'thedatavisible' or 'thedatair' nodes wrapped in individual 'observationalastronomy' nodes. So the results of four queries were put into four 'observationalastronomy' nodes in the XMLDocument, three queries into three and so on, as in the example below:

```
<?xml version="1.0" encoding="UTF-8"?>
- <astronomy xmlns:noNamespaceSchemaLocation="file:///C:/ROAD//RetrievalOfAstronomicalData//RetrievalOfAstronomicalData//astronomy.xsd">
  <astrophysics/>
  <astrobiology/>
  <astrochemistry/>
  <archeoastronomy/>
  <planetaryscience/>
  - <observationalastronomy>
    + <visible>
  </observationalastronomy>
  - <observationalastronomy>
    + <visible>
  </observationalastronomy>
  - <observationalastronomy>
    + <visible>
  </observationalastronomy>
  - <observationalastronomy>
    + <infrared>
  </observationalastronomy>
</astronomy>
```

This ensured that the data results returned were all stored in a distinct and separate manner. This allowed for output data files to be constructed, using further XSLT files, from a combination of this data according to the outputs that were desired by the user.

Due to the fact that output requirements from combined data sources can vary greatly, this ability to merge distinct structured data into XMLDocument output files enables better presentation and analysis of the data. This is because the data can be filtered and displayed at the point of output file creation from all the data received from all the datasources, no data or metadata being inadvertently 'dropped' along the

way. The consistent, defined structure of the XMLDocument holding the result data means that the complexity of producing the output files is reduced as there is a ‘baseline’ of data structure to work from, the XMLSchema validated XMLDocument.

7.4.2. How the XMLSchema handles the data and how it can be expanded

The structure of the XMLSchema ‘astronomy.xsd’ was based on the pattern ‘Venetian Blind’ as shown in Appendix 20 of this document. This was done to facilitate better the need for a single root element – that of ‘astronomy’, reusable complex types and reusable element groups. It was important that the design should be easily extensible, without the need to make changes to the existing structure hierarchy. For the purposes of this research this necessitated that the visible and infrared ‘observationalastronomy’ structure of the XMLSchema should be fully developed from top to bottom but that other areas could be left undeveloped at this time. During the research, therefore, the XMLSchema was developed to consist of the following structure:

```

Astronomy (root)
  Astrophysics
  Astrobiology
  Astrochemistry
  Archeoastronomy
  Planetary Science
  Observational Astronomy
    Visible
    Infrared
    Radio
    Microwave
    Gravity wave
    Shortwave
    Neutrino
    Submillimetre

```

All the branches of Astronomy other than ‘Observational Astronomy’ can be developed at some time subsequently as can all the types of ‘Observational Astronomy’ observations, other than ‘Visible’ and ‘Infrared’ types which were fully developed to the following detailed structure as shown below:

Visible		Infrared	
device	manufacturer	device	manufacturer
	model		model
location	gridref	location	gridref
	address		address
target	targetname	target	targetname
	targetposition		targetposition
targetephemeris	ephemeris	targetephemeris	ephemeris
datetime	thedata	datetime	thedata
	thetime		thetime
weather	weatherdesc	weather	weatherdesc
observer	forename	observer	forename
	surname		surname
visibleobservationdata	contact	irobservationdata	contact
	iscolour		thetime
	magnitude		theintensity
	description		thewavelength
	fileurlvisible		fileurlir
	filenamevisible		filenameeir
	thedatavisible		thedatair
	thedatavisibleblob (blob data)		thedatairblob (blob data)
images	filename	images	filename
	fileurl		fileurl
comments	imagedata (blob data)	comments	imagedata (blob data)
	thecomments		thecomments

The inclusion of blob data (proposed as Base 64 encoding) storage nodes is particularly valuable as it allows for streams of data to be added to the xml that it is not possible to add as 'string' data.

To allow for queries of multiple databases there is no limit to the number of instances of 'observationalastronomy' nodes in any given XMLDocument. This was shown during the research which produced XMLDocuments containing multiple 'observationalastronomy' nodes. There is also allowance in the XMLSchema for multiple nodes of 'Visible' and 'Infrared' observational types within an 'observationalastronomy' node, with no limit set there either.

A simplified structure of this schema can be seen in Appendix 1 whilst the XMLSchema text can be seen in Appendix 2 and by opening in the digital resource of the application source code the file 'astronomy.xsd'.

This schema is capable of further development without compromising the structure created so far. This is due to the fact that the root node is that of 'astronomy' with all the different areas of astronomy as child nodes. More child nodes can be added or existing ones further developed as needed.

7.4.3. The importance of schema validation

The whole point of validation was to test whether the XMLDocument produced was of the desired structure. For this research the system application was constructed so that it would only produce output files for the user if the XMLDocument was successfully validated. The section of code carrying out this validation was implemented in the road.aspx.cs class (Appendix 9) and is shown below:

```
// NOTE: This validation code (somewhat altered) taken originally from: https://msdn.microsoft.com/en-GB/library/bb387037.aspx on 24-02-2015
XmlSchemaSet schemas = new XmlSchemaSet();
schemas.Add("", "C://ROAD//RetrievalOfAstronomicalData//RetrievalOfAstronomicalData//RetrievalOfAstronomicalData//astronomy.xsd");
XDocument xdocResults = XDocument.Parse(resultsXmlDoc.OuterXml);

bool errors = false;
xdocResults.Validate(schemas, (o, ve) => {
    sw.WriteLine("{0}", ve.Message);
    errors = true;
});

sw.WriteLine("resultsXmlDoc {0}", errors ? "did not validate" : "validated");
sw.WriteLine();
```

You can see in the code that the XMLDocument ‘resultsXmlDoc’ that is produced from the chosen datasources is parsed against the XMLSchema ‘astronomy.xsd’.

If the validation fails and the code boolean object ‘errors’ is therefore true the specific error of the failure is written to the log file via the StreamWriter ‘sw’. Only if ‘errors’ is false are any user output documents produced, as shown in the code below:

```
if (!errors) {
// IF VALIDATED, OUTPUT THE DATA IN CHOSEN FILE TYPES - ELSE output Error to log file
    chosenOutputs = new SortedList();
    IEnumerator outChoicesEnum = lstSelectedOutputs.Items.GetEnumerator();
    while (outChoicesEnum.MoveNext()) {
        ListItem outChoicesItem = (ListItem)outChoicesEnum.Current;
        chosenOutputs.Add(outChoicesItem.Text, outChoicesItem.Value);
    }

    createChosenOutputs = new OutputTransform();
    createChosenOutputs.createOutputs(chosenOutputs, resultsXmlDoc, sw);

    sw.WriteLine("");
    sw.WriteLine("Outputs produced");
    txtLog.Text = txtLog.Text + ("") + "\n";
    txtLog.Text = txtLog.Text + ("Outputs produced") + "\n";
}
```

Validation makes the whole business of data retrieval and its use more standardized and controllable. This standardization of the retrieval of astronomical data has benefits in reducing the time, effort and cost of using stored astronomical data.

To be able to use a widely accepted XMLSchema would only increase these benefits further. As has been pointed out in the literature review earlier, there is a huge and ever increasing amount of astronomical data available and awaiting investigation. (Wall, 2014)

7.5. Suggestions for improvements

It quickly became apparent that there were improvements to the ROAD application that would be obviously beneficial such as those listed in Table 15 below:

Table 15: Suggested improvements to ROAD

Data that needs to be essential/mandatory	This could lead on to software requesting manual input into the data query results content if it is missing (the system could demand correct data or an insertion of some kind of explanatory notes)
Security	Possible benefits of security and XML encryption could also be considered
Use of more observation types	ie: more than just Visible and Infra-red
Unrestricted choice of datasources	
Unrestricted choice of astronomical targets	
Build up a library of XSLT files	for datasource query data transformation... and to a lesser extent for transforms creating output files
Build up a library of query produced XMLDocuments	for general use
Consultation with astronomical research and industry sectors	
W3C accreditation	Work towards developing fully and achieving W3C accreditation for the astronomy.xsd XMLSchema

Implementation of these improvements would produce a more beneficial ROAD application. This amount of work is however, way beyond the scope of this master's research project.

7.6. The importance of the research

It has been shown from this research that a data structure standard can be created in the form of an XMLSchema so that queries of the extremely varied and disparate astronomical databases are returned (and stored) in a common format of XMLDocument.

Building up a generally available library Of the XMLDocuments from completed database queries would

be of great benefit for accessing data. Also, building up an available library of XSLT files to be used for querying astronomical datasources would be an increasingly valuable asset for the creation of XMLDocuments containing astronomical data.

In the same way, for generating outputs of various types, another library of XSLT files would be of importance also.

Whilst pointing the way forwards, the results from this research only serve to show how much more needs to be done if a useable, widely accepted XMLSchema is to be created.

Contact with the astronomical community would be an essential part of any work that is intended to further develop the astronomical XMLSchema. This would also be true in order to carry out the creation of a substantial XSLT library. This is due to the fact that understanding the needs and wishes of astronomical organizations would be complex and time consuming research requiring both technical understanding and substantial 'people skills'.

7.7. Barriers and limitations

The first immediate barrier that became apparent upon commencement of system development was one of finding suitable datasources. The reticence of the astronomical research and industry sectors to providing any source of data was a surprise. Fortunately it was possible to find some suitable publically available sources online. Although not ideal, they just about provided a wide enough range of data types for this proof of concept to show it can process varied data types. The astronomical data types queried from databases during this research are shown in Table 16 below:

Table 16: Table of datasource data types

Datasource	Datatype	Datatype Description
MAST	CSV	Delimited flat file
OEC_XML	XML	Structured markup file
OEC	HTML	Webpage content
SIMBAD	HTML	Webpage content

Another barrier was a personal lack of knowledge in some XML technologies, notably XSL-FO. This was anticipated however and some time was spent learning the skills required. Nevertheless the XSL-FO skills remained pretty basic. This can be seen by looking at the PDF outputs produced, in that the layouts are at

best mediocre. Despite this they are adequate for the purpose in that they display data from multiple sources, filtered and brought together into a single PDF document. This can be seen in Appendix 6 and by opening in the digital resource the output file ‘run1output.pdf’.

It was discovered that XSLT could not be made to carry out all the transform actions that were expected, particularly in the more sophisticated manipulation of string data. The barriers arose in the output transforms when creating the layouts of documents for both HTML and PDF outputs. These manipulations had to be carried out in the C# code of the application before and after the use of the XSLT output files. This can be seen in the source code of the application in the method ‘outputPdf’ of the ‘OutputTransform’ class, some of which is shown here:

```
// Declare svg fully for xsl-fo transform
if (pdfTransformStr.Contains("<svg>")) {
    pdfTransformStr = pdfTransformStr.Replace("<svg xmlns:svg=\"http://www.w3.org/2000/svg\" width=\"480\" height=\"400\">", "</fo:block><fo:block><fo:instream-foreign-object></fo:block><fo:block font-size=\"10pt\" font-family=\"<g style=\"stroke:black;\">", "<svg:g style=\"stroke:black;\">");
    pdfTransformStr = pdfTransformStr.Replace("<g>", "<svg:g>");
    pdfTransformStr = pdfTransformStr.Replace("</g>", "</svg:g>");
    pdfTransformStr = pdfTransformStr.Replace("<text>", "<svg:text>");
    pdfTransformStr = pdfTransformStr.Replace("</text>", "</svg:text>");
    pdfTransformStr = pdfTransformStr.Replace("<circle>", "<svg:circle>");
    pdfTransformStr = pdfTransformStr.Replace("<line>", "<svg:line>");
    pdfTransformStr = pdfTransformStr.Replace("style=\"fill:url(#g2); stroke:none\"", "fill=\"lightgrey\"");
    pdfTransformStr = pdfTransformStr.Replace("style=\"fill:url(#g3); stroke:none\"", "fill=\"brown\"");
    pdfTransformStr = pdfTransformStr.Replace("ellipse", "svg:ellipse");
    pdfTransformStr = pdfTransformStr.Replace("<svg xmlns:svg=\"http://www.w3.org/2000/svg\" width=\"600\" height=\"100\">", "</fo:block><fo:block><fo:instream-foreign-object></fo:block><fo:block font-size=\"10pt\" font-family=\"<g style=\"stroke:black;\">", "<svg:g style=\"stroke:black;\">");
    pdfTransformStr = pdfTransformStr.Replace("</fo:block></fo:flow></fo:page-sequence></fo:root>", "</fo:block><fo:flow></fo:page-sequence></fo:root>");
    pdfTransformStr = pdfTransformStr.Replace("<p>", "");
    pdfTransformStr = pdfTransformStr.Replace("</p>", "");
    pdfTransformStr = pdfTransformStr.Replace("<h3>", "");
    pdfTransformStr = pdfTransformStr.Replace("</h3>", "");
    pdfTransformStr = pdfTransformStr.Replace("fill=\"url(#habitablegradient)\"", "fill=\"lightgreen\"");
    pdfTransformStr = pdfTransformStr.Replace("<g style=\"stroke:black;\">", "<svg:g style=\"stroke:black;\">");
}
```

It remains to be seen whether this type of string manipulation could be achieved within XSLT with greater knowledge and expertise of XML Transforms.

The research embarked upon here was strictly and deliberately limited in its scope because of the fact that master’s research gives a relatively short period of time in which to produce a viable piece of work out of what is a potentially huge subject area. In order to complete this work it was decided to construct a working application to demonstrate the technologies being discussed, with limited available datasources, no choice of target and basic user interface and output requirements.

7.8. Implications and Impacts

This demonstration by the application proof of concept shows a way of retrieval that still leaves the original data residing in all the original varied databases. Whilst being able to go back to the original data storage has the benefits of ‘going to the source’ one can begin to see, as a natural

consequence of the transformational methods used, another repository that can be developed from all this disparate source data.

If a standard XMLSchema for data structure can be accepted then a number of beneficial tools can be developed. A library of XMLDocuments could be built up for accessing information, the data being in the schema validated structure of these documents. XMLDocuments could be added to the library every time a query is made to astronomical databases. Additionally a library of XSLT files can also be built up in a similar way, in that every time a new XSLT file is created it can be added to the XSLT library for people to use.

In this way a source of astronomical data with a common structure can be 'grown' by everyone who uses it from all areas of astronomy. If there are restricted items of data that are not to be accessed by everyone, this data can be protected in the XMLDocuments by using the security and encryption methods available within XML Technologies. Also, essential metadata items can be made mandatory by the software requiring content to be added if it is missing.

The implications and beneficial impacts of having such a structured resource of astronomical data, organically growing through use are several.

There would be benefits of cheaper data retrieval due to greater efficiency, fewer transformational steps than some currently require and reduced metadata loss. A benefit also would be the creation and continual expansion of an XMLDocument data library. Additionally an XSLT library for the storage of the files and their associated datasource queries used in the transformation of data source query results can be set up along with another XSLT library for storing and using output file transforms.

Continual additions to these libraries would mean an ever growing resource of XMLDocuments, XSLT files and datasource queries.

Indeed, over time, once the resource is properly established, it may be that querying multiple numbers of XMLDocuments from the resource may become a frequent source of data for many users.

Chapter 8

Conclusions

8.1. Key Findings

This research achieved the aims and objectives stated in Chapter 1. A way was looked for to retrieve data in a more efficient manner from all the many varied and disparate ways that astronomical data has been saved over the years. The purpose was to show that the data could be retrieved and then be presented in a standard format for easier and more economical use. This was done by:

- Implementing the proof of concept (which made use of XML Technologies) and carrying out the test runs that not only retrieved data but also produced validated XMLDocuments holding the retrieved data, subject to successful validation by the XMLSchema.
- The correct operation of XMLDocument against XMLSchema validation was also demonstrated during the test runs.
- In addition some variations of output files were produced from the XMLDocuments. Although basic in layout these outputs showed the technical operation involved in producing an end product (output files) from the data held in the XMLDocuments.
- The implementation of the software system also made clear that XSLT files were a requirement both for the creation of XMLDocuments from source data and the production of output files for the end user.

It is this XMLSchema validation process that provides the control on XMLDocument structure to ensure a standard structure. It is this standardization of structure that can reduce the loss of metadata (by ensuring that it is included) and enable easier retrieval and use.

Using a resource of this type would help to interpret, analyse and evaluate the huge backlog of stored astronomical data and the new data being stored.

8.2. Future Research

The implications in section 7.8 of this document clearly indicate that future research is required for the development of such a resource. Such research would require skills of both a technical and

sociological nature. On the technical side the XMLSchema needs to be extended across the full spectrum of astronomical disciplines and more XSLT files and data queries need to be developed. On the sociological side, this solution to the data retrieval problem needs to be attractive to the astronomical research and industry sectors. Achieving the W3C standard for the XML Schema would be a big step forward in achieving widespread acceptance.

It would be necessary to go out to the various astronomical organizations to get their opinions and professional input. Only with their interest and engagement would it be possible to bring to reality a resource that would be truly beneficial to the task of astronomical data retrieval. This indicates that cross-over research with industry is a possibility.

9 Appendices

Appendix 1

Astronomy.xsd structure

Astronomy (root)

```

  Astrophysics
  Astrobiology
  Astrochemistry
  Archeoastronomy
  Planetary Science
  Observational Astronomy
  Visible
    device
      manufacturer
      model
    location
      gridref
      address
    target
      targetname
      targetposition
    targetephemeris
      ephemeris
    datetime
      thedate
      thetime
    weather
      weatherdesc
    observer
      forename
      surname
      contact
    visibleobservationdata
      iscolour
      magnitude
      description
      fileurlvisible
      filenamevisible
      thedatavisible
      thedatavisibleblob (blob data)
    images
      filename
      fileurl
      imagedata (blob data)
    comments
      thecomments
  Infrared
    device
      manufacturer
      model
    location
      gridref
      address
    target
      targetname
      targetposition
    targetephemeris
      ephemeris

```



```

datetime
    thedate
    thetime
weather
    weatherdesc
observer
    forename
    surname
    contact
iobservationdata
    thetime
    theintensity
    thewavelength
    fileurlir
    filenameeir
    thedatair
    thedatairblob (blob data)
images
    filename
    fileurl
    imagedata (blob data)
comments
    thecomments
Radio
Microwave
Gravity wave
Shortwave
Neutrino
Submillimetre

```

Appendix 2 – astronomy.xsd

```

<?xml version = "1.0" ?>
<xs:schema xmlns:xs = "http://www.w3.org/2001/XMLSchema">

  <!-- NOTES: This schema follows the Venetian Blind pattern -->
  <!-- For large data - images, tables etc... propose to use the Base 64 encoding approach -->
  <!-- http://www.javaworld.com/article/2077481/learn-java/java-tip-117-transfer-binary-data-in-an-xml-document.html -->

  <!-- GENERAL COMPLEX TYPES -->
  <!-- ===== -->

  <xs:complexType name="device.type">
    <xs:sequence>
      <xs:element name="manufacturer" type="xs:string"/>
      <xs:element name="model" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="location.type">
    <xs:sequence>
      <xs:element name="gridRef" type="xs:string"/>
      <xs:element name="address" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="target.type">
    <xs:sequence>
      <xs:element name="targetname" type="xs:string"/>
      <xs:element name="targetposition" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="targetephemeris.type">
    <xs:sequence>
      <xs:element name="ephemeris" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="datetime.type">
    <xs:sequence>
      <xs:element name="thedata" type="xs:string"/>
      <xs:element name="thetime" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="weather.type">
    <xs:sequence>
      <xs:element name="weatherdesc" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="observer.type">
    <xs:sequence>
      <xs:element name="forename" type="xs:string"/>
      <xs:element name="surname" type="xs:string"/>
      <xs:element name="contact" type="xs:string"/>
    </xs:sequence>

```

```

</xs:complexType>

<xs:complexType name="visibleobservationdata.type">
  <xs:sequence>
    <xs:element name="iscolour" type="xs:string"/>
    <xs:element name="magnitude" type="xs:string"/>
    <xs:element name="description" type="xs:string"/>
    <xs:element name="fileurlvisible" type="xs:string"/>
    <xs:element name="filenamevisible" type="xs:string"/>
    <xs:element name="thedatavisible" type="xs:string"/>
    <xs:element name="thedatavisibleblob" type="xs:string"/> <!-- blob type Base 64 encoding -->
  </xs:sequence>
</xs:complexType>

<xs:complexType name="irobservationdata.type">
  <xs:sequence>
    <xs:element name="thetime" type="xs:string"/>
    <xs:element name="theintensity" type="xs:string"/>
    <xs:element name="thewavelength" type="xs:string"/>
    <xs:element name="description" type="xs:string"/>
    <xs:element name="fileurlir" type="xs:string"/>
    <xs:element name="filenameir" type="xs:string"/>
    <xs:element name="thedatair" type="xs:string"/>
    <xs:element name="thedatairblob" type="xs:string"/> <!-- blob type Base 64 encoding -->
  </xs:sequence>
</xs:complexType>

<xs:complexType name="images.type">
  <xs:sequence>
    <xs:element name="fileurl" type="xs:string"/>
    <xs:element name="filename" type="xs:string"/>
    <xs:element name="imagedata" type="xs:string"/> <!-- blob type Base 64 encoding -->
  </xs:sequence>
</xs:complexType>

<xs:complexType name="comments.type">
  <xs:sequence>
    <xs:element name="thecomments" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

<!-- CHILDREN OF OBSERVATIONALASTRONOMY NODE -->
<!-- ===== -->
<xs:complexType name="visible.type">
  <xs:sequence>
    <xs:element name='device' type="device.type"/>
    <xs:element name='location' type="location.type"/>
    <xs:element name='target' type="target.type"/>
    <xs:element name='targetephemeris' type="targetephemeris.type"/>
    <xs:element name='datetime' type="datetime.type"/>
    <xs:element name='weather' type="weather.type"/>
    <xs:element name='observer' type="observer.type"/>
    <xs:element name='visibleobservationdata' type="visibleobservationdata.type"/>
    <xs:element name='images' type="images.type"/>
    <xs:element name='comments' type="comments.type"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="radio.type">

```

```

<xs:sequence>
  <!-- *** TODO *** -->
</xs:sequence>
</xs:complexType>

<xs:complexType name="infrared.type">
  <xs:sequence>
    <xs:element name='device' type="device.type"/>
    <xs:element name='location' type="location.type"/>
    <xs:element name='target' type="target.type"/>
    <xs:element name='targetephemeris' type="targetephemeris.type"/>
    <xs:element name='datetime' type="datetime.type"/>
    <xs:element name='weather' type="weather.type"/>
    <xs:element name='observer' type="observer.type"/>
    <xs:element name='irobservationdata' type="irobservationdata.type"/>
    <xs:element name='images' type="images.type"/>
    <xs:element name='comments' type="comments.type"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="microwave.type">
  <xs:sequence>
    <!-- *** TODO *** -->
  </xs:sequence>
</xs:complexType>

<xs:complexType name="gravitywave.type">
  <xs:sequence>
    <!-- *** TODO *** -->
  </xs:sequence>
</xs:complexType>

<xs:complexType name="shortwave.type">
  <xs:sequence>
    <!-- *** TODO *** -->
  </xs:sequence>
</xs:complexType>

<xs:complexType name="neutrino.type">
  <xs:sequence>
    <!-- *** TODO *** -->
  </xs:sequence>
</xs:complexType>

<xs:complexType name="submillimetre.type">
  <xs:sequence>
    <!-- *** TODO *** -->
  </xs:sequence>
</xs:complexType>

<!-- CHILDREN OF ROOT NODE -->
<!-- ===== -->
<xs:complexType name="observationalastronomy.type">
  <xs:choice>
    <xs:element name='visible' type="visible.type" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name='radio' type="radio.type" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name='infrared' type="infrared.type" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name='microwave' type="microwave.type" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name='gravitywave' type="gravitywave.type" minOccurs="0" maxOccurs="unbounded"/>
  </xs:choice>

```

```

    <xs:element name='shortwave' type="shortwave.type" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name='neutrino' type="neutrino.type" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name='submillimetre' type="submillimetre.type" minOccurs="0" maxOccurs="unbounded"/>
  </xs:choice>
</xs:complexType>

<xs:complexType name="astrophysics.type">
  <xs:sequence>
    <!-- *** TODO *** -->
  </xs:sequence>
</xs:complexType>

<xs:complexType name="astrobiology.type">
  <xs:sequence>
    <!-- *** TODO *** -->
  </xs:sequence>
</xs:complexType>

<xs:complexType name="astrochemistry.type">
  <xs:sequence>
    <!-- *** TODO *** -->
  </xs:sequence>
</xs:complexType>

<xs:complexType name="archeoastronomy.type">
  <xs:sequence>
    <!-- *** TODO *** -->
  </xs:sequence>
</xs:complexType>

<xs:complexType name="planetaryscience.type">
  <xs:sequence>
    <!-- *** TODO *** -->
  </xs:sequence>
</xs:complexType>

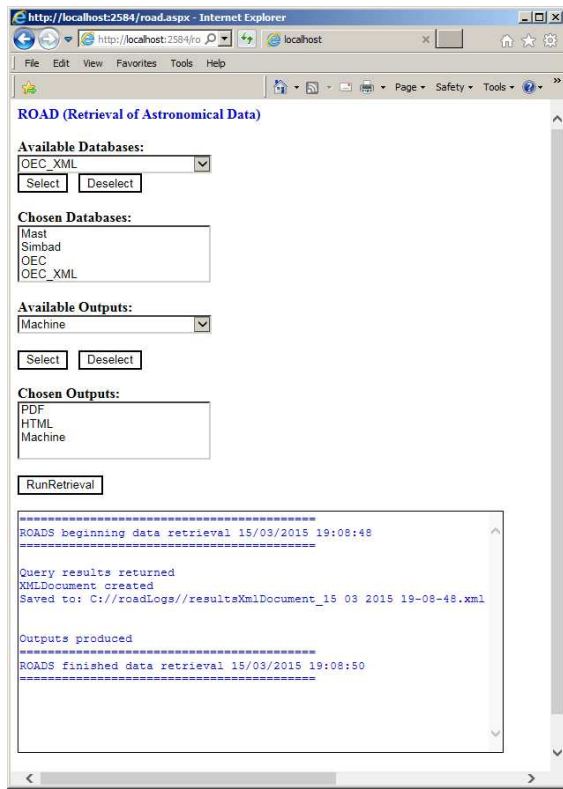
<!-- ROOT NODE -->
<!-- ===== -->
<xs:element name="astronomy">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="astrophysics" type="astrophysics.type" maxOccurs="unbounded"/>
      <xs:element name="astrobiology" type="astrobiology.type" maxOccurs="unbounded"/>
      <xs:element name="astrochemistry" type="astrochemistry.type" maxOccurs="unbounded"/>
      <xs:element name="archeoastronomy" type="archeoastronomy.type" maxOccurs="unbounded"/>
      <xs:element name="planetaryscience" type="planetaryscience.type" maxOccurs="unbounded"/>
      <xs:element name="observationalastronomy" type="observationalastronomy.type" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

</xs:schema>

```

Appendix 3

Run 1



```
=====
ROADS beginning data retrieval 15/03/2015 19:08:48
=====
```

```
Querying for data...
```

```
Querying for Mast
```

```
Querying for OEC
```

```
Querying for OEC_XML
```

```
Querying for Simbad
```

```
Query results returned
```

```
resultsXmlDoc validated
```

```
XMLDocument created: C://roadLogs//resultsXmlDocument_15 03 2015 19-08-48.xml
```

```
Getting output choices...
```

```
Choosing HTML
```

```
htmlOutput created: C://roadLogs//outputs//htmlOutput_15 03 2015 19-08-50.html
```

```
Choosing Machine
```

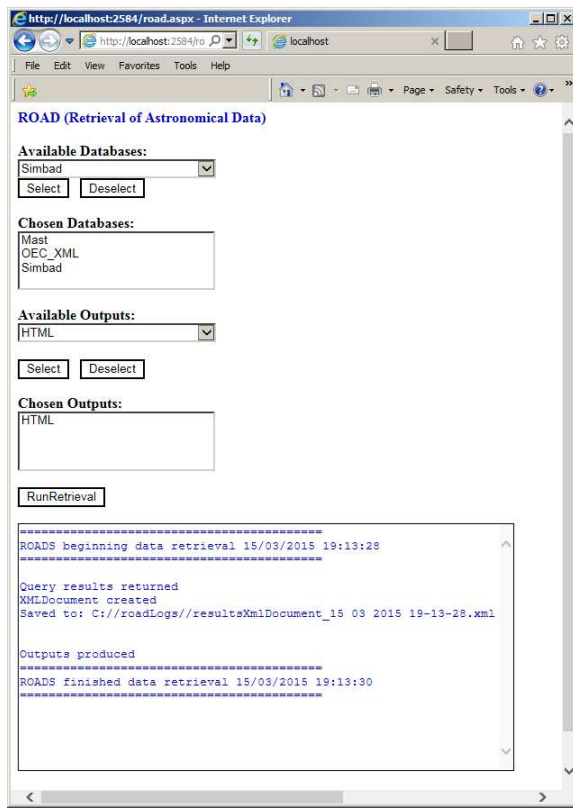
```
machineOutput created: C://roadLogs//outputs//machineOutput_15 03 2015 19-08-50.xml
```

```
Choosing PDF
```

```
pdfOutput created: C://roadLogs//outputs//pdfOutput_15 03 2015 19-08-50.fo
```

```
Outputs produced
```

```
=====
ROADS finished data retrieval 15/03/2015 19:08:50
=====
```

Run 2

```
=====
ROADS beginning data retrieval 15/03/2015 19:13:28
=====
```

```
Querying for data...
```

```
Querying for Mast
```

```
Querying for OEC_XML
```

```
Querying for Simbad
```

```
Query results returned
```

```
resultsXmlDoc validated
```

```
XMLDocument created: C://roadLogs//resultsXmlDocument_15 03 2015 19-13-28.xml
```

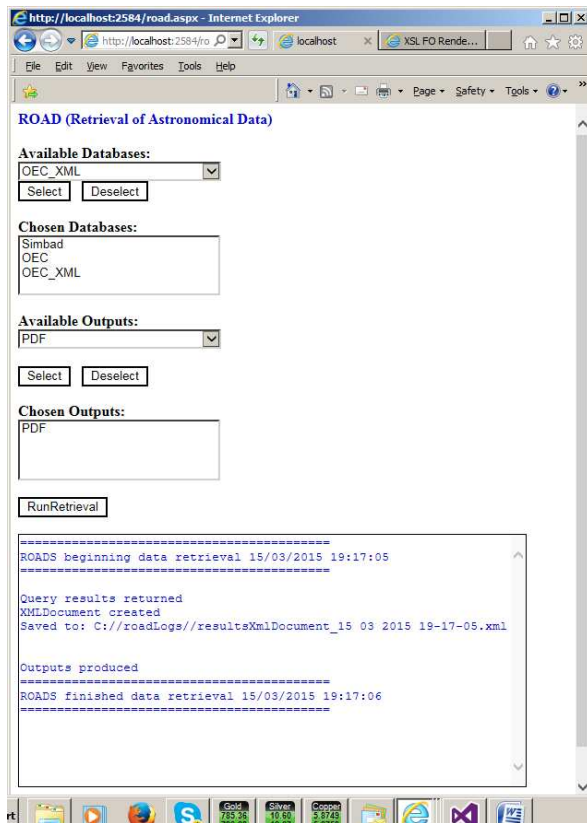
```
Getting output choices...
```

```
Choosing HTML
```

```
htmlOutput created: C://roadLogs//outputs//htmlOutput_15 03 2015 19-13-30.html
```

```
Outputs produced
```

```
=====
ROADS finished data retrieval 15/03/2015 19:13:30
=====
```

Run 3

```

=====
ROADS beginning data retrieval 15/03/2015 19:17:05
=====

```

```

Querying for data...

```

```

Querying for OEC
Querying for OEC_XML
Querying for Simbad

```

```

Query results returned
resultsXmlDoc validated

```

```

XMLDocument created: C://roadLogs//resultsXmlDocument_15 03 2015 19-17-05.xml

```

```

Getting output choices...

```

```

Choosing PDF
pdfOutput created: C://roadLogs//outputs//pdfOutput_15 03 2015 19-17-06.fo

```

```

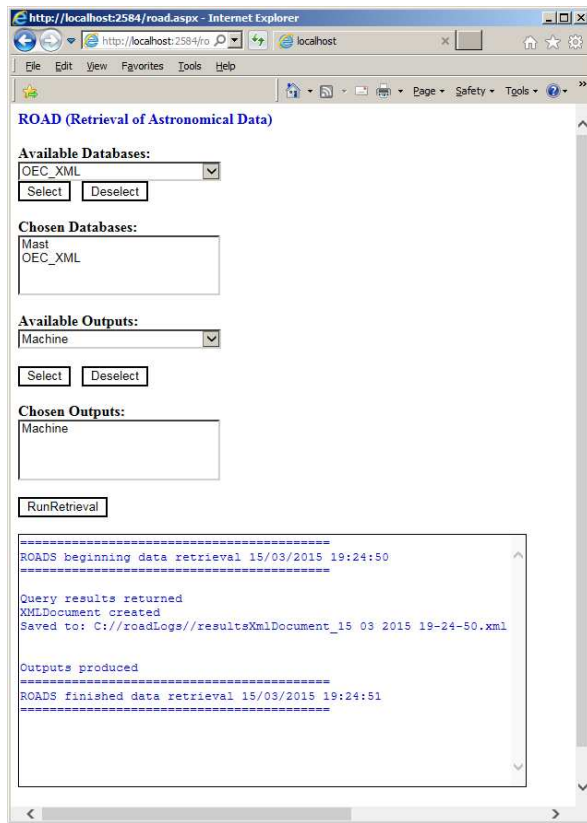
Outputs produced

```

```

=====
ROADS finished data retrieval 15/03/2015 19:17:06
=====

```


Run 4

```

=====
ROADS beginning data retrieval 15/03/2015 19:24:50
=====

```

```

Querying for data...

```

```

Querying for Mast
Querying for OEC_XML

```

```

Query results returned
resultsXmlDoc validated

```

```

XMLDocument created: C://roadLogs//resultsXmlDocument_15 03 2015 19-24-50.xml

```

```

Getting output choices...

```

```

Choosing Machine
machineOutput created: C://roadLogs//outputs//machineOutput_15 03 2015 19-24-51.xml

```

```

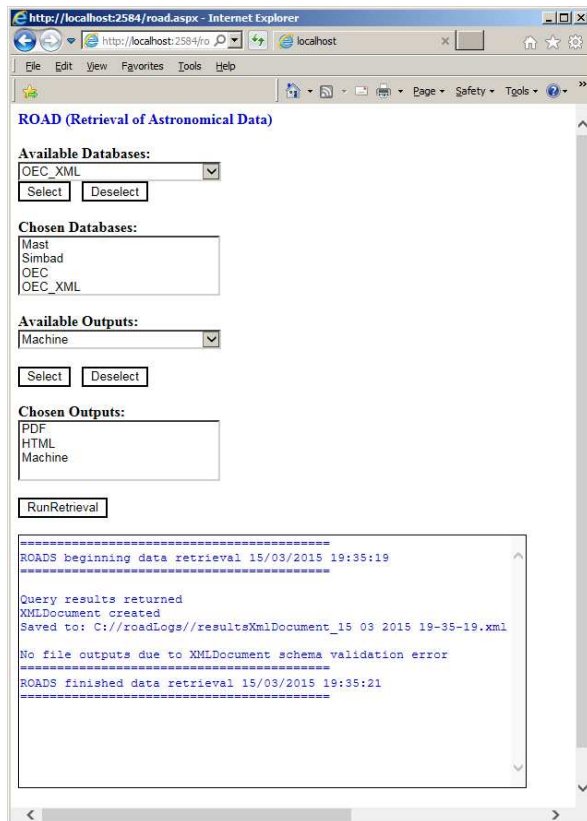
Outputs produced

```

```

=====
ROADS finished data retrieval 15/03/2015 19:24:51
=====

```

Run 5

```
=====
ROADS beginning data retrieval 15/03/2015 19:35:19
=====
```

```
Querying for data...
```

```
Querying for Mast
Querying for OEC
Querying for OEC_XML
Querying for Simbad
```

```
Query results returned
The element 'infrared' has invalid child element 'visualobservationdata'. List of possible elements expected: 'irobservationdata'.
resultsXmlDoc did not validate
```

```
XMLDocument created: C://roadLogs//resultsXmlDocument_15 03 2015 19-35-19.xml
```

```
*** Error *** No file outputs due to XMLDocument schema validation error
```

```
=====
ROADS finished data retrieval 15/03/2015 19:35:21
=====
```

Appendix 4

The validated XMLDocument ‘resultsXmlDocument_15 03 2015 19-08-48.xml’

```
<?xml version="1.0" encoding="utf-8"?>
<astronomy xmlns:noNamespaceSchemaLocation="file:///C:/ROAD/RetrievalOfAstronomicalData/RetrievalOfAstronomicalData/astronomy.xsd">
  <astrophysics />
  <astrobiology />
  <astrochemistry />
  <archeoastronomy />
  <planetaryscience />
  <observationalastronomy>
    <visible>
      <device>
        <manufacturer />
        <model>SIMBAD</model>
      </device>
      <location>
        <gridRef />
        <address>Strasbourg</address>
      </location>
      <target>
        <targetname>Tau Ceti</targetname>
        <targetposition />
      </target>
      <targetephemeris>
        <ephemeris>
          </ephemeris>
        </targetephemeris>
      <datetime>
        <thedata>15 03 2015</thedata>
        <thetime />
      </datetime>
      <weather>
        <weatherdesc>
          </weatherdesc>
        </weather>
      <observer>
        <forename />
        <surname>
          </surname>
        <contact />
      </observer>
      <visibleobservationdata>
        <iscolour />
        <magnitude />
        <description>SIMBADHTML</description>
        <fileurlvisible />
        <filenamevisible />
        <thedatavisible>
          ![CDATA["http://simbad.u-strasbg.fr/simbad/sim-id?ident=%401298185&Name=%20tau%20Cet&submit=submit"]]
        </thedatavisible>
        <thedatavisibleblob />
      </visibleobservationdata>
    </visible>
  </observationalastronomy>
</astronomy>
```

```

</visibleobservationdata>
<images>
  <fileurl>
  </fileurl>
  <filename />
  <imagedata />
</images>
<comments>
  <thecomments>For furthur info check the data</thecomments>
</comments>
</visible>
</observationalastronomy>
<observationalastronomy>
  <visible>
    <device>
      <manufacturer />
      <model>OEC</model>
    </device>
    <location>
      <gridRef />
      <address>GITHUB</address>
    </location>
    <target>
      <targetname>Tau Ceti</targetname>
      <targetposition />
    </target>
    <targetephemeris>
      <ephemeris>
      </ephemeris>
    </targetephemeris>
    <datetime>
      <thedata>15 03 2015</thedata>
      <thetime />
    </datetime>
    <weather>
      <weatherdesc>
      </weatherdesc>
    </weather>
    <observer>
      <forename />
      <surname>
      </surname>
      <contact />
    </observer>
  </visibleobservationdata>
  <iscolour />
  <magnitude />
  <description>XML</description>
  <fileurivisible />
  <filenamevisible />
  <thedatavisible>
    &lt;system&gt;

```

<name>tau Ceti</name>
 <name>HD 10700</name>
 <rightascension>01 44 04</rightascension>
 <declination>-15 56 15</declination>
 <distance>3.65</distance>
 <star>
 <magB>4.22</magB>
 <magV>3.50</magV>
 <magJ errorminus="0.31" errorplus="0.31">2.15</magJ>
 <magH errorminus="0.234" errorplus="0.234">1.800</magH>
 <magK errorminus="0.274" errorplus="0.274">1.794</magK>
 <name>tau Ceti</name>
 <name>HD 10700</name>
 <mass>0.783</mass>
 <radius>0.798</radius>
 <metallicity>-0.55</metallicity>
 <age>5.8</age>
 <spectralttype>G8.5 V</spectralttype>
 <planet>
 <name>tau Ceti b</name>
 <name>HD 10700 b</name>
 <list>Controversial</list>
 <mass>0.0062914013</mass>
 <period>13.965</period>
 <semimajoraxis>0.105</semimajoraxis>
 <eccentricity>0.16</eccentricity>
 <discoverymethod>RV</discoverymethod>
 <lastupdate>12/12/19</lastupdate>
 <discoveryyear>2012</discoveryyear>
 <description>Multiple periodic signals were found by Tuomi et al. in 2012 in the radial velocity data from HARPS, AAPS and HIRES. The planetary nature of the signals has yet to be confirmed.</description>
 <image>tau_ceti</image>
 <imagedescription>Artist's impression of the Tau Ceti system

Picture credit: J. Pinfield for the RoPACS network at the University of Hertfordshire, 2012.</imagedescription>

<planet>
 <planet>
 <name>tau Ceti c</name>
 <name>HD 10700 c</name>
 <list>Controversial</list>
 <mass>0.009751672</mass>
 <period>35.362</period>
 <semimajoraxis>0.195</semimajoraxis>
 <eccentricity>0.03</eccentricity>
 <discoverymethod>RV</discoverymethod>
 <lastupdate>12/12/19</lastupdate>
 <discoveryyear>2012</discoveryyear>
 <description>Multiple periodic signals were found by Tuomi et al. in 2012 in the radial velocity data from HARPS, AAPS and HIRES. The planetary nature of the signals has yet to be confirmed.</description>
 <image>tau_ceti</image>
 <imagedescription>Artist's impression of the Tau Ceti system

Picture credit: J. Pinfield for the RoPACS network at the University of Hertfordshire, 2012.

planet

planet

name:tau Ceti

name:HD 10700

list:Controversial

mass:0.011324522

period:94.11

semimajoraxis:0.374

eccentricity:0.08

discoverymethod:RV

lastupdate:12/12/19

discoveryyear:2012

description:Multiple periodic signals were found by Tuomi et al. in 2012 in the radial velocity data from HARPS, AAPS and HIRES. The planetary nature of the signals has yet to be confirmed.

image:tau_ceti

imagedescription:Artist's impression of the Tau Ceti system

Picture credit: J. Pinfield for the RoPACS network at the University of Hertfordshire, 2012.

planet

planet

name:tau Ceti e

name:HD 10700 e

list:Controversial

mass:0.013526513

period:168.12

semimajoraxis:0.552

eccentricity:0.085

discoverymethod:RV

lastupdate:12/12/19

discoveryyear:2012

description:Multiple periodic signals were found by Tuomi et al. in 2012 in the radial velocity data from HARPS, AAPS and HIRES. The planetary nature of the signals has yet to be confirmed.

image:tau_ceti

imagedescription:Artist's impression of the Tau Ceti system

Picture credit: J. Pinfield for the RoPACS network at the University of Hertfordshire, 2012.

planet

planet

name:tau Ceti f

name:HD 10700 f

list:Controversial

mass:0.020761624

period:642

semimajoraxis:1.35

eccentricity:0.03

discoverymethod:RV

lastupdate:12/12/19

discoveryyear:2012

description:Multiple periodic signals were found by Tuomi et al. in 2012 in the radial velocity data from HARPS, AAPS and HIRES. The planetary nature of the signals has yet to be confirmed.

image:tau_ceti

<imagedescription>Artist's impression of the Tau Ceti system

Picture credit: J. Pinfield for the RoPACS network at the University of Hertfordshire, 2012.</imagedescription>

</planet>

</star>

</system>

</thedatavisible>

<thedatavisibleblob />

</visibleobservationdata>

<images>

<fileurl>

</fileurl>

<filename />

<imagedata />

</images>

<comments>

<thecomments>For furthur info check the data</thecomments>

</comments>

</visible>

</observationalastronomy>

<observationalastronomy>

<visible>

<device>

<manufacturer />

<model>OEC</model>

</device>

<location>

<gridRef />

<address>GITHUB</address>

</location>

<target>

<targetname>Tau Ceti</targetname>

<targetposition />

</target>

<targetephemeris>

<ephemeris>

</ephemeris>

</targetephemeris>

<datetime>

<thedata>15 03 2015</thedata>

<thetime />

</datetime>

<weather>

<weatherdesc>

</weatherdesc>

</weather>

<observer>

<forename />

<surname>

</surname>

<contact />

```

</observer>

<visibleobservationdata>

<iscolour />

<magnitude />

<description>HTML</description>

<fileurlvisible />

<filenamevisible />

<thedatavisible>

    ![CDATA[&lt;h3&gt;Planet sizes&lt;/h3&gt;

&lt;p&gt;

The following plot shows the approximate sizes of the planets in this system

The Solar System planets are shown as a comparison.

Note that unless the radius has been determined through a transit observation, this is only an approximation (see Lissauer et al. 2011b).

&lt;/p&gt;

&lt;svg width="480" height="400"&gt;

    &lt;defs&gt;

        &lt;radialGradient id = "g1" cx = "50%" cy = "50%" r = "50%"&gt;

            &lt;stop style="stop-color:rgb(20,20,20);" offset = "0%"&gt;

            &lt;stop style="stop-color:rgb(100,100,100);" offset = "95%"&gt;

            &lt;stop style="stop-color:rgb(250,250,250);" offset = "100%"&gt;

        &lt;/radialGradient&gt;

        &lt;radialGradient id = "g2" cx = "50%" cy = "50%" r = "50%"&gt;

            &lt;stop style="stop-color:rgb(120,120,120);" offset = "0%"&gt;

            &lt;stop style="stop-color:rgb(180,180,180);" offset = "95%"&gt;

            &lt;stop style="stop-color:rgb(252,252,252);" offset = "100%"&gt;

        &lt;/radialGradient&gt;

        &lt;radialGradient id = "g3" cx = "50%" cy = "50%" r = "50%"&gt;

            &lt;stop style="stop-color:rgb(120,80,80);" offset = "0%"&gt;

            &lt;stop style="stop-color:rgb(180,100,100);" offset = "95%"&gt;

            &lt;stop style="stop-color:rgb(252,202,202);" offset = "100%"&gt;

        &lt;/radialGradient&gt;

    &lt;/defs&gt;

    &lt;g style="stroke:black;"&gt;

        &lt;circle cx="2.102509" cy="300.000000" r="1.102509" style="fill:url(#g2); stroke:none" /&gt;&lt;line x1="2.000000" y1="209.000000" x2="2.000000" y2="300.000000" fill="none" stroke="lightgrey"

        /&gt;&lt;text x="2.000000" y="215.000000" font-family="sans-serif" font-weight="normal" font-size="12.000000" stroke="none" &gt;Pluto&lt;/text&gt;&lt;circle cx="17.523137" cy="300.000000" r="2.318120"

        style="fill:url(#g2); stroke:none" /&gt;&lt;line x1="16.205018" y1="224.000000" x2="16.205018" y2="300.000000" fill="none" stroke="lightgrey" /&gt;&lt;text x="16.205018" y="230.000000" font-family="sans-

        serif" font-weight="normal" font-size="12.000000" stroke="none" &gt;Mercury&lt;/text&gt;&lt;circle cx="35.069897" cy="300.000000" r="3.228640" style="fill:url(#g2); stroke:none" /&gt;&lt;line

        x1="32.841257" y1="239.000000" x2="32.841257" y2="300.000000" fill="none" stroke="lightgrey" /&gt;&lt;text x="32.841257" y="245.000000" font-family="sans-serif" font-weight="normal" font-

        size="12.000000" stroke="none" &gt;Mars&lt;/text&gt;&lt;circle cx="56.050591" cy="300.000000" r="5.752054" style="fill:url(#g2); stroke:none" /&gt;&lt;line x1="51.298537" y1="254.000000"

        x2="51.298537" y2="300.000000" fill="none" stroke="lightgrey" /&gt;&lt;text x="51.298537" y="260.000000" font-family="sans-serif" font-weight="normal" font-size="12.000000" stroke="none"

        &gt;Venus&lt;/text&gt;&lt;circle cx="79.864543" cy="300.000000" r="6.061897" style="fill:url(#g2); stroke:none" /&gt;&lt;line x1="74.802646" y1="269.000000" x2="74.802646" y2="300.000000" fill="none"

        stroke="lightgrey" /&gt;&lt;text x="74.802646" y="275.000000" font-family="sans-serif" font-weight="normal" font-size="12.000000" stroke="none" &gt;Earth&lt;/text&gt;&lt;circle cx="121.943074"

        cy="300.000000" r="24.016633" style="fill:url(#g2); stroke:none" /&gt;&lt;line x1="98.926441" y1="284.000000" x2="98.926441" y2="300.000000" fill="none" stroke="lightgrey" /&gt;&lt;text x="98.926441"

        y="290.000000" font-family="sans-serif" font-weight="normal" font-size="12.000000" stroke="none" &gt;Neptune&lt;/text&gt;&lt;circle cx="182.251968" cy="300.000000" r="24.292260" style="fill:url(#g2);

        stroke:none" /&gt;&lt;line x1="158.959707" y1="299.000000" x2="158.959707" y2="300.000000" fill="none" stroke="lightgrey" /&gt;&lt;text x="158.959707" y="305.000000" font-family="sans-serif" font-

        weight="normal" font-size="12.000000" stroke="none" &gt;Uranus&lt;/text&gt;&lt;circle cx="275.825262" cy="300.000000" r="57.281034" style="fill:url(#g2); stroke:none" /&gt;&lt;line x1="219.544228"

        y1="314.000000" x2="219.544228" y2="300.000000" fill="none" stroke="lightgrey" /&gt;&lt;text x="219.544228" y="320.000000" font-family="sans-serif" font-weight="normal" font-size="12.000000"

        stroke="none" &gt;Saturn&lt;/text&gt;&lt;circle cx="413.053153" cy="300.000000" r="67.946856" style="fill:url(#g2); stroke:none" /&gt;&lt;line x1="346.106296" y1="329.000000" x2="346.106296"

        y2="300.000000" fill="none" stroke="lightgrey" /&gt;&lt;text x="346.106296" y="335.000000" font-family="sans-serif" font-weight="normal" font-size="12.000000" stroke="none"

        &gt;Jupiter&lt;/text&gt;&lt;circle cx="9.668873" cy="100.000000" r="8.668873" style="fill:url(#g3); stroke:none" /&gt;&lt;line x1="2.000000" y1="9.000000" x2="2.000000" y2="100.000000" fill="none"

        stroke="lightgrey" /&gt;&lt;text x="2.000000" y="15.000000" font-family="sans-serif" font-weight="normal" font-size="12.000000" stroke="none" &gt;tau Ceti b&lt;/text&gt;&lt;circle cx="41.061741"

```



```

cy="100.000000" r="10.723996" style="fill:url(#g3); stroke:none" />&lt;
line x1="31.337745" y1="24.000000" x2="31.337745" y2="100.000000" fill="none" stroke="lightgrey" />&lt;
text x="31.337745" y="30.000000" font-family="sans-serif" font-weight="normal" font-size="12.000000" stroke="none" &lt;
tau Ceti c&lt;/text&gt;&lt;
circle cx="75.317118" cy="100.000000" r="11.531381" style="fill:url(#g3); stroke:none" />&lt;
line x1="64.785738" y1="39.000000" x2="64.785738" y2="100.000000" fill="none" stroke="lightgrey" />&lt;
text x="64.785738" y="45.000000" font-family="sans-serif" font-weight="normal" font-size="12.000000" stroke="none" &lt;
tau Ceti d&lt;/text&gt;&lt;
circle cx="111.418651" cy="100.000000" r="12.570152" style="fill:url(#g3); stroke:none" />&lt;
line x1="99.848499" y1="54.000000" x2="99.848499" y2="100.000000" fill="none" stroke="lightgrey" />&lt;
text x="99.848499" y="60.000000" font-family="sans-serif" font-weight="normal" font-size="12.000000" stroke="none" &lt;
tau Ceti e&lt;/text&gt;&lt;
circle cx="151.465148" cy="100.000000" r="15.476345" style="fill:url(#g3); stroke:none" />&lt;
line x1="136.988802" y1="69.000000" x2="136.988802" y2="100.000000" fill="none" stroke="lightgrey" />&lt;
text x="136.988802" y="75.000000" font-family="sans-serif" font-weight="normal" font-size="12.000000" stroke="none" &lt;
tau Ceti f&lt;/text&gt;&lt;
/g&gt;&lt;
/svg&gt;

```

<h3>Habitable zone</h3>

<p>

The following plot shows the approximate location of the planets in this system with respect to the habitable zone (green) and the size of the star (red).

This is only an estimate, using the star's spectral type and mass.

Note that if no green band is shown in the plot, then the planet's orbit is far outside the habitable zone.

The equations of <http://adsabs.harvard.edu/abs/2007A%26A...476.1373S> & Selsis, Kasting et al & are used to draw the inner and outer boundaries.

<p>

<svg width="600" height="100">

```

&lt;defs&gt;
  &lt;radialGradient id="habitablegradient" &gt;
    &lt;stop id="stops0" offset="0" stop-color="lightgreen" stop-opacity="0"/&gt;
    &lt;stop id="stops1" offset="0.343789" stop-color="lightgreen" stop-opacity="0"/&gt;
    &lt;stop id="stops2" offset="0.480324" stop-color="lightgreen" stop-opacity="1"/&gt;
    &lt;stop id="stops3" offset="0.858408" stop-color="lightgreen" stop-opacity="1"/&gt;
    &lt;stop id="stops4" offset="1" stop-color="lightgreen" stop-opacity="0"/&gt;
  &lt;/radialGradient&gt;
&lt;/defs&gt;

&lt;ellipse cx="0.000000" cy="50.000000" rx="440.053612" ry="440.053612" fill="url(#habitablegradient)" />&lt;
text x="151.285662" y="99.000000" font-family="sans-serif" font-weight="normal" font-size="12" stroke="none" style="fill:green"&gt;Habitable zone&lt;/text&gt;&lt;
ellipse cx="0.000000" cy="50.000000" rx="1.347563" ry="1.347563" style="fill:red" />&lt;
style="stroke:black"/&gt;&lt;
g&gt;&lt;
ellipse cx="0.000000" cy="50.000000" rx="38.111111" ry="38.111111" style="fill:none" />&lt;
text x="40.111111" y="24.000000" font-family="sans-serif" font-weight="normal" font-size="12.000000" stroke="none" &lt;
tau Ceti b&lt;/text&gt;&lt;
g&gt;&lt;
g&gt;&lt;
ellipse cx="0.000000" cy="50.000000" rx="70.777778" ry="70.777778" style="fill:none" />&lt;
text x="72.777778" y="36.000000" font-family="sans-serif" font-weight="normal" font-size="12.000000" stroke="none" &lt;
tau Ceti c&lt;/text&gt;&lt;
g&gt;&lt;
g&gt;&lt;
ellipse cx="0.000000" cy="50.000000" rx="135.748148" ry="135.748148" style="fill:none" />&lt;
text x="137.748148" y="48.000000" font-family="sans-serif" font-weight="normal" font-size="12.000000" stroke="none" &lt;
tau Ceti d&lt;/text&gt;&lt;
g&gt;&lt;
g&gt;&lt;
ellipse cx="0.000000" cy="50.000000" rx="200.355556" ry="200.355556" style="fill:none" />&lt;
text x="202.355556" y="60.000000" font-family="sans-serif" font-weight="normal" font-size="12.000000" stroke="none" &lt;
tau Ceti e&lt;/text&gt;&lt;
g&gt;&lt;
g&gt;&lt;
ellipse cx="0.000000" cy="50.000000" rx="490.000000" ry="490.000000" style="fill:none" />&lt;
text x="492.000000" y="72.000000" font-family="sans-serif" font-weight="normal" font-size="12.000000" stroke="none" &lt;
tau Ceti f&lt;/text&gt;&lt;
/g&gt;&lt;
/svg&gt;
]]

```

```

</thedatavisible>

<thedatavisibleblob />

</visibleobservationdata>

<images>

<fileurl>

</fileurl>

<filename />

<imagedata />

</images>

<comments>

<thecomments>For further info check the data</thecomments>

```

```

</comments>

</visible>

</observationalastronomy>

<observationalastronomy>

<infrared>

<device>

  <manufacturer />

  <model>MAST Repository</model>

</device>

<location>

  <gridRef />

  <address>USA</address>

</location>

<target>

  <targetname>Tau Ceti</targetname>

  <targetposition />

</target>

<targetephemeris>

  <ephemeris>

  </ephemeris>

</targetephemeris>

<datetime>

  <thedata>15 03 2015</thedata>

  <thetime />

</datetime>

<weather>

  <weatherdesc>

  </weatherdesc>

</weather>

<observer>

  <forename />

  <surname>

  </surname>

  <contact />

</observer>

<irobservationdata>

  <thetime />

  <theintensity />

  <thewavelength />

  <description>CSV</description>

  <fileurlir />

  <filenameeir />

  <thedatair>Dataset, Target Name, RA (J2000), Dec (J2000), Ref, Start Time, Stop Time, Exp Time, Instrument, Apertures, Filters/Gratings, Central Wavelength, Proposal ID, Release Date, Preview
Name, High-Level Science Products, Ang Sep (')

string, string, ra, dec, integer, datetime, datetime, float, string, string, string, float, integer, datetime, string, integer, float

U33LC701M, CL-ULIR00060-1543, 00 08 35.281, -15 27 20.22, 35, 1997-05-07 07:53:14, 1997-05-07 07:59:54, 400.000, WFPC2 , WF3, F814W, 8012.000, 6346, 1998-05-07 13:13:50, U33LC701M,, 41.353

U33LC702M, CL-ULIR00060-1543, 00 08 35.281, -15 27 20.22, 35, 1997-05-07 08:02:14, 1997-05-07 08:08:54, 400.000, WFPC2 , WF3, F814W, 8012.000, 6346, 1998-05-07 13:06:10, U33LC702M,, 41.353

N52XAKRRQ, ANY, 00 02 12.868, -15 32 28.25, 2, 1998-09-26 01:54:44, 1998-09-26 02:11:48, 1023.950, NICMOS, NIC2, F222M, 22181.699, 8082, 1998-09-26 18:27:26, N52XAKRRQ,, 51.248

N4XFGESUQ, ANY, 00 02 12.868, -15 32 28.25, 1, 1998-09-26 06:32:14, 1998-09-26 06:39:42, 447.976, NICMOS, NIC2, F222M, 22181.699, 8063, 1998-09-26 17:33:24, N4XFGESUQ,, 51.248

N4XFGT3Q, ANY, 00 02 12.868, -15 32 28.25, 1, 1998-09-26 06:51:39, 1998-09-26 06:59:07, 447.976, NICMOS, NIC2, F222M, 22181.699, 8063, 1998-09-26 18:06:37, N4XFGT3Q,, 51.248

N52XALS5Q, ANY, 00 02 12.868, -15 32 28.25, 2, 1998-09-26 03:30:25, 1998-09-26 03:47:29, 1023.950, NICMOS, NIC2, F222M, 22181.699, 8082, 1998-09-26 18:42:19, N52XALS5Q,, 51.248

N52XAMSJQ, ANY, 00 02 12.868, -15 32 28.25, 2, 1998-09-26 05:07:13, 1998-09-26 05:24:17, 1023.950, NICMOS, NIC2, F222M, 22181.699, 8082, 1998-09-26 18:49:49, N52XAMSJQ,, 51.248

```

N52XAKRJQ,ANY,00 02 12.780,-15 32 29.77,2,1998-09-26 01:34:28,1998-09-26 01:51:32,1023.950,NICMOS,NIC2,F222M,22181.699,8082,1998-09-26 15:54:00,N52XAKRJQ,,51.269
N52XALRXQ,ANY,00 02 12.780,-15 32 29.77,2,1998-09-26 03:10:09,1998-09-26 03:27:13,1023.950,NICMOS,NIC2,F222M,22181.699,8082,1998-09-26 16:31:04,N52XALRXQ,,51.269
N52XAMSDQ,ANY,00 02 12.780,-15 32 29.77,2,1998-09-26 04:46:57,1998-09-26 05:04:01,1023.950,NICMOS,NIC2,F222M,22181.699,8082,1998-09-26 17:09:52,N52XAMSDQ,,51.269
</thedatair>
 <thedatairblob />
</iobservationdata>
<images>
 <fileurl>
 </fileurl>
 <filename />
 <imagedata />
</images>
<comments>
 <thecomments>For furthur info check the data</thecomments>
</comments>
</infrared>
</observationalastronomy>
</astronomy>

Appendix 5

Web page produced in Run 1 'htmlOutput_15 03 2015 19-08-50.html'

Tau Ceti

STELLA (Resolving astrometrical Data Center)

Take this link for STERAD star data and visual image provided by Aladin Lite (sky atlas) [Tau Ceti](#)

ORC_XML (Open Right Ascension Catalogue - XML, 64K)

Name: tau Ceti Right Ascension: 01:41:00 Declination: -12:56:18 Distance: 3.65

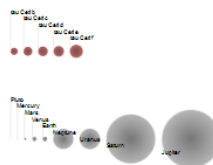
Planets of tau Ceti

Name	Lat	Mass	Period	Semi Major Axis	Discovery Method	Last Update	Discovery Year	Description
tau Ceti b	Conventional	0.0040916112	13.660	0.105	RV	12/12/19	2012	Multiple periodic signals were found by Tsiou et al. in 2012 in the radial velocity data from HARPS, AAPS and HRES. The planetary nature of the signals has yet to be confirmed.
tau Ceti c	Conventional	0.00971875	34.560	0.195	RV	12/12/19	2012	Multiple periodic signals were found by Tsiou et al. in 2012 in the radial velocity data from HARPS, AAPS and HRES. The planetary nature of the signals has yet to be confirmed.
tau Ceti d	Conventional	0.011216522	94.11	0.274	RV	12/12/19	2012	Multiple periodic signals were found by Tsiou et al. in 2012 in the radial velocity data from HARPS, AAPS and HRES. The planetary nature of the signals has yet to be confirmed.
tau Ceti e	Conventional	0.015256512	148.12	0.352	RV	12/12/19	2012	Multiple periodic signals were found by Tsiou et al. in 2012 in the radial velocity data from HARPS, AAPS and HRES. The planetary nature of the signals has yet to be confirmed.
tau Ceti f	Conventional	0.020761624	645	1.32	RV	12/12/19	2012	Multiple periodic signals were found by Tsiou et al. in 2012 in the radial velocity data from HARPS, AAPS and HRES. The planetary nature of the signals has yet to be confirmed.

ORC (Open Right Ascension Catalogue)

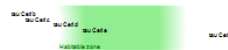
Planet sizes

The following plot shows the approximate size of the planets in this system. The Solar System planets are shown as a comparison. Note that unless the radius has been determined through a transit observation, this is only an approximation (see Lemaire et al. 2013).



Habitable zone

The following plot shows the approximate location of the planets in this system with respect to the habitable zone (green). This is only an estimate, using the star's spectral type and mass. Note that if the green band is shown in the plot, then the planets orbit is far outside the habitable zone. The equations of [Kopke, Kratochvil et al.](#) were used to draw the inner and outer boundaries.

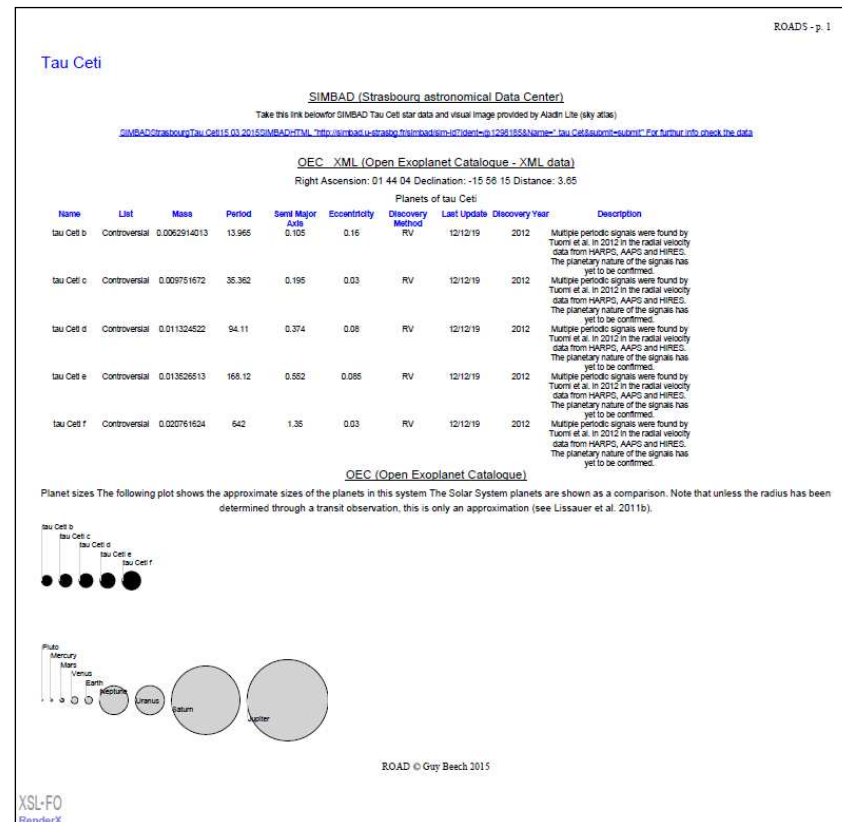


From the MAST Repository (IR data - csv)

Planet	Target Name	RA (J2000)	Dec (J2000)	Ref	Start Time	Stop Time	Exp Time	Instrument	Aperture	Filter	Grating	Central Wavelength	Proposed ID	Release Date	Preview Name	High-Level Science Products	Log Sep (")
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:41:00	0.000000	WFPCC1	WFP	F1147W		8012.000	0246	1999-05-07	12 06 10	U1375462	01.202
U1375462	CL-LULIR0000-1602	00 18 10.281	-12 57 10.22 18	1997-05-07	01:41:00	01:4											

Appendix 6

PDF produced in Run 1 'run1output.pdf'



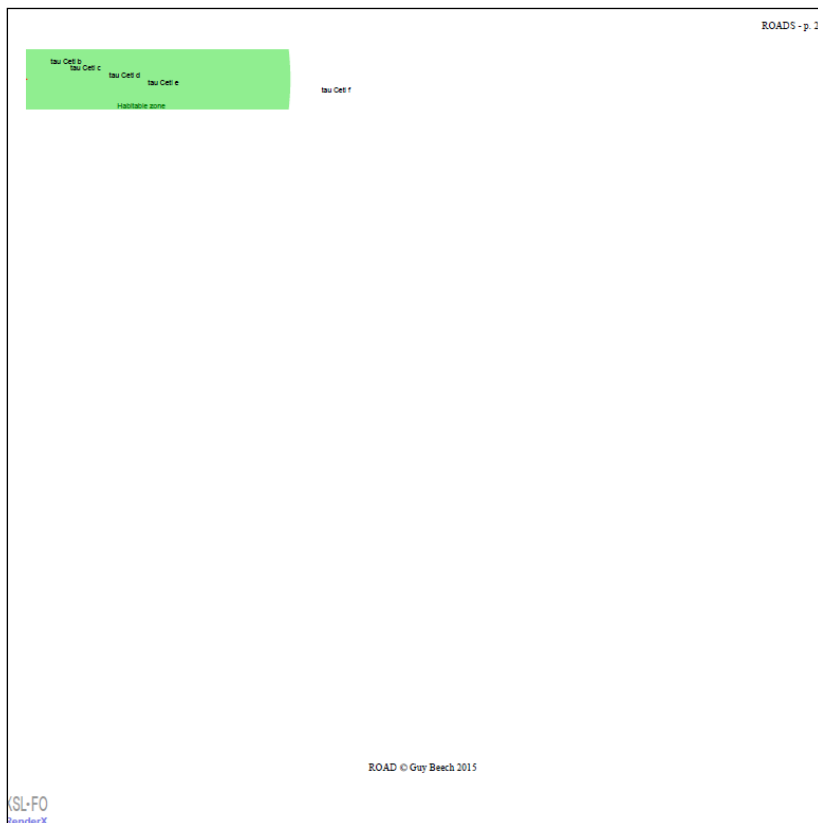
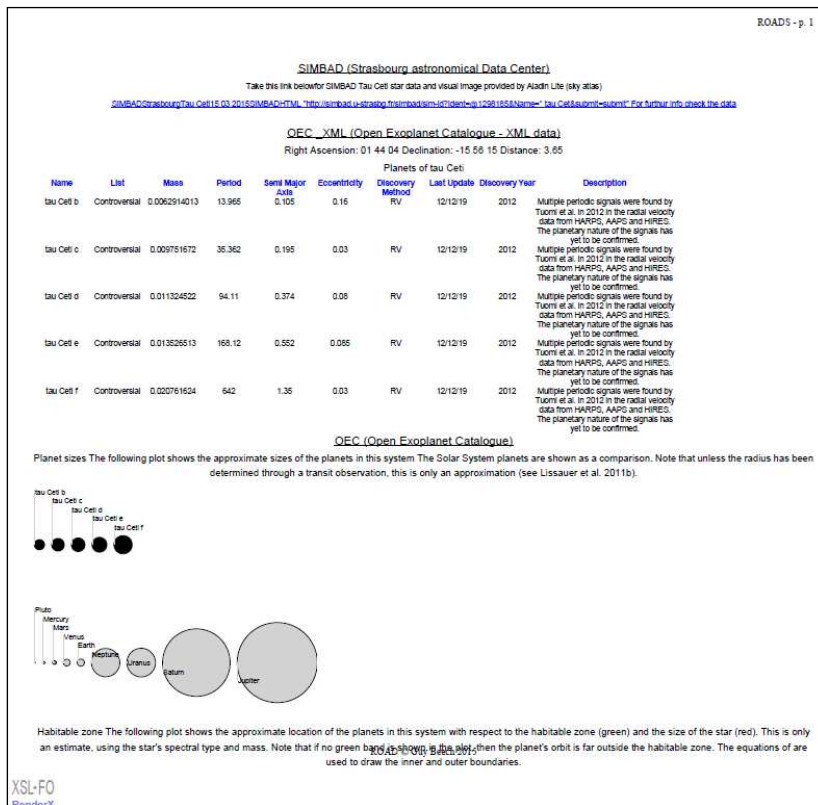
Appendix 7

Web page produced in Run 2 'htmlOutput_15 03 2015 19-13-30.html'

Tau Ceti																
<div>STIRAD (Strasbourg astronomical Data Center)</div> <div>Take this link for STIRAD star data and visual image provided by Aladin Lite (sky atlas) Tau Ceti</div>																
<div>OEC XML (Open Exoplanet Catalogue - XML data)</div> <div>Name: tau Ceti Right Ascension: 01 44 04 Declination: -15 56 15 Distance: 3.65</div>																
Planets of tau Ceti																
Name	List	Mass	Period	Semi-Major Axis	Eccentricity	Discovery Method	Last Update	Discovery Year	Description							
tau Ceti b	Conversational	0.0062914013	13.965	0.105	0.16	RV	12/12/19	2012	Multiple periodic signals were found by Tuomi et al. in 2012 in the radial velocity data from HARPS, AAPS and HIRES. The planetary nature of the signals has yet to be confirmed.							
tau Ceti c	Conversational	0.009751672	35.362	0.195	0.03	RV	12/12/19	2012	Multiple periodic signals were found by Tuomi et al. in 2012 in the radial velocity data from HARPS, AAPS and HIRES. The planetary nature of the signals has yet to be confirmed.							
tau Ceti d	Conversational	0.011324522	94.11	0.374	0.08	RV	12/12/19	2012	Multiple periodic signals were found by Tuomi et al. in 2012 in the radial velocity data from HARPS, AAPS and HIRES. The planetary nature of the signals has yet to be confirmed.							
tau Ceti e	Conversational	0.013526513	168.12	0.552	0.085	RV	12/12/19	2012	Multiple periodic signals were found by Tuomi et al. in 2012 in the radial velocity data from HARPS, AAPS and HIRES. The planetary nature of the signals has yet to be confirmed.							
tau Ceti f	Conversational	0.020761624	642	1.35	0.03	RV	12/12/19	2012	Multiple periodic signals were found by Tuomi et al. in 2012 in the radial velocity data from HARPS, AAPS and HIRES. The planetary nature of the signals has yet to be confirmed.							
From the MAST Repository (IR data - csv)																
Dataset	Target Name	RA (J2000)	Dec (J2000)	Ref	Start Time	Stop Time	Exp Time	Instrument	Apertures	Filters/Gratings	Central Wavelength	Proposal ID	Release Date	Preview Name	High-Level Science Products	Ang Sep (")
U33LC701M	CL-ULIR00000-1540	00 08 -15 27	35 281	20 22	35 1997-05-07	1997-05-07	400.000	WFPIC2	WF3	F814W	8012.000	6346	1998-05-07 13:13:50	U33LC701M		41.353
U33LC702M	CL-ULIR00000-1540	00 08 -15 27	35 281	20 22	35 1997-05-07	1997-05-07	400.000	WFPIC2	WF3	F814W	8012.000	6346	1998-05-07 13:06:10	U33LC702M		41.353
N52XAKRQ	ANY	00 02 -15 32	12.868	28.25	2	1998-09-26 01:54:44	02:11:48	1023.950	NICMOS	NIC2	F222M	22181.699	8082	1998-09-26 18:27:26	N52XAKRQ	51.248
N4XFGESUQ	ANY	00 02 -15 32	12.868	28.25	1	1998-09-26 06:32:14	06:39:42	447.976	NICMOS	NIC2	F222M	22181.699	8063	1998-09-26 17:33:24	N4XFGESUQ	51.248
N4XFGFTIQ	ANY	00 02 -15 32	12.868	28.25	1	1998-09-26 06:51:39	06:59:07	447.976	NICMOS	NIC2	F222M	22181.699	8063	1998-09-26 18:06:37	N4XFGFTIQ	51.248
N52XAL55Q	ANY	00 02 -15 32	12.868	28.25	2	1998-09-26 03:30:25	03:47:29	1023.950	NICMOS	NIC2	F222M	22181.699	8082	1998-09-26 18:42:19	N52XAL55Q	51.248
N52XAMISQ	ANY	00 02 -15 32	12.868	28.25	2	1998-09-26 05:07:13	05:24:17	1023.950	NICMOS	NIC2	F222M	22181.699	8082	1998-09-26 18:49:49	N52XAMISQ	51.248
N52XAKRQ	ANY	00 02 -15 32	12.780	29.77	2	1998-09-26 01:34:28	01:51:32	1023.950	NICMOS	NIC2	F222M	22181.699	8082	1998-09-26 15:54:00	N52XAKRQ	51.269
N52XALRXQ	ANY	00 02 -15 32	12.780	29.77	2	1998-09-26 03:10:09	03:27:13	1023.950	NICMOS	NIC2	F222M	22181.699	8082	1998-09-26 16:51:04	N52XALRXQ	51.269
N52XAMISQ	ANY	00 02 -15 32	12.780	29.77	2	1998-09-26 04:46:57	05:04:01	1023.950	NICMOS	NIC2	F222M	22181.699	8082	1998-09-26 17:09:52	N52XAMISQ	51.269
ROADS Â Guy Beech 2015																

Appendix 8

PDF produced in Run 3 'run3output.pdf'



Appendix 9 - road.aspx.cs

```

using System;
using System.Collections;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Configuration;
using System.IO;
using System.Xml;
using System.Xml.Schema;
using System.Xml.XPath;
using System.Xml.Linq;

namespace RetrievalOfAstronomicalData {
    public partial class road : System.Web.UI.Page {
        // CENTRALISED VARIABLES
        private SortedList chosenDatasources;
        private SortedList queryResults;
        private SortedList chosenOutputs;
        private Query dataQueries;
        private StreamWriter sw;
        private XmlDocument resultsXmlDoc;
        private QueryTransform transformResults;
        private OutputTransform createChosenOutputs;

        protected void Page_Load(object sender, EventArgs e) {
            // DATASOURCE CHOICES FOR THE USER
            if (ddlDatabases.Items.Count == 0) {
                ddlDatabases.Items.Add(new ListItem("Mast", "http://archive.stsci.edu/hst/search.php?RA=01.44&DEC=-15.56&radius=100.&max_records=10&outputformat=CSV&action=Search"));
                ddlDatabases.Items.Add(new ListItem("Simbad", "http://simbad.u-strasbg.fr/simbad/sim-coo?output.format=HTML&Coord=01 44 -15 56&Radius=10&Radius.unit=arcmin"));
                ddlDatabases.Items.Add(new ListItem("OEC", "http://www.openexoplanetcatalogue.com/planet/tau%20Ceti%20d/"));
                ddlDatabases.Items.Add(new ListItem("OEC_XML", "http://www.openexoplanetcatalogue.com/open_exoplanet_catalogue/systems/tau%20Ceti.xml"));
            }
            queryResults = new SortedList();

            // OUTPUT CHOICES FOR THE USER
            if (ddlChosenOutputs.Items.Count == 0) {
                ddlChosenOutputs.Items.Add(new ListItem("PDF", "PDF"));
                ddlChosenOutputs.Items.Add(new ListItem("HTML", "HTML"));
                ddlChosenOutputs.Items.Add(new ListItem("Machine", "Machine"));
            }
            chosenOutputs = new SortedList();
        }

        /// <summary>
        /// Adds selected database to the list of selected databases
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        protected void btnSelectDatabase_Click(object sender, EventArgs e) {
            ListItem lstItem = new ListItem(ddlDatabases.SelectedItem.Text, ddlDatabases.SelectedItem.Value);
            if (!lstSelectedDatabases.Items.Contains(lstItem)) {
                lstSelectedDatabases.Items.Add(lstItem);
            }
        }

        /// <summary>
        /// Removes selected database from the list of selected databases
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        protected void btnDeselectDatabase_Click(object sender, EventArgs e) {
            ListItem lstItem = new ListItem(ddlDatabases.SelectedItem.Text, ddlDatabases.SelectedItem.Value);
            lstSelectedDatabases.Items.Remove(lstItem);
        }
    }
}

```



```

}

/// <summary>
/// Removes selected output from the list of selected outputs
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void btnDeselectOutput_Click(object sender, EventArgs e) {
    ListItem lstItem = new ListItem(ddlChosenOutputs.SelectedItem.Text, ddlChosenOutputs.SelectedItem.Value);
    lstSelectedOutputs.Items.Remove(lstItem);
}

/// <summary>
/// Adds selected database to the list of selected outputs
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void btnSelectOutput_Click(object sender, EventArgs e) {
    ListItem lstItem = new ListItem(ddlChosenOutputs.SelectedItem.Text, ddlChosenOutputs.SelectedItem.Value);
    if (!lstSelectedOutputs.Items.Contains(lstItem)) {
        lstSelectedOutputs.Items.Add(lstItem);
    }
}

/// <summary>
/// Retrieves data from the list of selected databases
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void btnRetrieval_Click(object sender, EventArgs e) {
    txtLog.Text = "";
    // CREATE NEW TEXT FILE EACH TIME PROCESS IS RUN
    string timeStr = DateTime.Now.ToString();
    timeStr = timeStr.Replace("/", " ");
    timeStr = timeStr.Replace(":", "-");
    // Incorporate date time into file name
    string path = "C://roadLogs//myTest_" + timeStr + ".txt";
    // Create new folder if not exists.
    if (Directory.Exists("C://roadLogs") == false) {
        DirectoryInfo dir = Directory.CreateDirectory("C://roadLogs");
    }
    // Create a new stream to write to the text log file
    sw = File.AppendText(path);
    sw.WriteLine("=====");
    sw.WriteLine("ROADS beginning data retrieval " + DateTime.Now);
    sw.WriteLine("=====");
    sw.Flush();
    txtLog.Text = txtLog.Text + ("=====") + "\n";
    txtLog.Text = txtLog.Text + ("ROADS beginning data retrieval " + DateTime.Now) + "\n";
    txtLog.Text = txtLog.Text + ("=====") + "\n";

    // GET DATA
    chosenDatasources = new SortedList();
    dataQueries = new Query();
    IEnumerable datasourcesEnum = lstSelectedDatabases.Items.GetEnumerator();
    while (datasourcesEnum.MoveNext()) {
        ListItem datasourcesItem = (ListItem)datasourcesEnum.Current;
        chosenDatasources.Add(datasourcesItem.Text, datasourcesItem.Value);
    }
    queryResults = dataQueries.queryAllSelectedDatasources(chosenDatasources, sw);
    sw.WriteLine("");
    sw.WriteLine("Query results returned");
    txtLog.Text = txtLog.Text + ("") + "\n";
    txtLog.Text = txtLog.Text + ("Query results returned") + "\n";

    // CREATE THE XML DOCUMENT AND TEST FOR SCHEMA COMPLIANCE
    transformResults = new QueryTransform();
    resultsXmlDoc = transformResults.transformIntoXmlDoc(queryResults, sw);
    txtLog.Text = txtLog.Text + ("XMLDocument created") + "\n";

```

```

//
=====
// NOTE: This validation code (somewhat altered) taken originally from: https://msdn.microsoft.com/en-GB/library/bb387037.aspx on 24-
0202015
//
=====
=====
XmlSchemaSet schemas = new XmlSchemaSet();
schemas.Add("",
"C://ROAD//RetrievalOfAstronomicalData//RetrievalOfAstronomicalData//RetrievalOfAstronomicalData//astronomy.xsd");
XDocument xdocResults = XDocument.Parse(resultsXmlDoc.OuterXml);

bool errors = false;
xdocResults.Validate(schemas, (o, ve) => {
    sw.WriteLine("{0}", ve.Message);
    errors = true;
});

sw.WriteLine("resultsXmlDoc {0}", errors ? "did not validate" : "validated");
sw.WriteLine();
//
=====
=====

// Output XmlDocument to file
resultsXmlDoc.Save("C://roadLogs//resultsXmlDocument_" + timeStr + ".xml");
sw.WriteLine("");
sw.WriteLine("XMLDocument created: " + "C://roadLogs//resultsXmlDocument_" + timeStr + ".xml");
txtLog.Text = txtLog.Text + ("Saved to: C://roadLogs//resultsXmlDocument_" + timeStr + ".xml") + "\n";
txtLog.Text = txtLog.Text + ("" + "\n";

if (!errors) {
    // IF VALIDATED, OUTPUT THE DATA IN CHOSEN FILE TYPES - ELSE output Error to log file
    chosenOutputs = new SortedList();
    IEnumerator outChoicesEnum = lstSelectedOutputs.Items.GetEnumerator();
    while (outChoicesEnum.MoveNext()) {
        ListItem outChoicesItem = (ListItem)outChoicesEnum.Current;
        chosenOutputs.Add(outChoicesItem.Text, outChoicesItem.Value);
    }

    createChosenOutputs = new OutputTransform();
    createChosenOutputs.createOutputs(chosenOutputs, resultsXmlDoc, sw);

    sw.WriteLine("");
    sw.WriteLine("Outputs produced");
    txtLog.Text = txtLog.Text + ("" + "\n";
    txtLog.Text = txtLog.Text + ("Outputs produced") + "\n";
}
else {
    sw.WriteLine("");
    sw.WriteLine("*** Error *** No file outputs due to XmlDocument schema validation error");
    txtLog.Text = txtLog.Text + ("No file outputs due to XmlDocument schema validation error");
    txtLog.Text = txtLog.Text + ("" + "\n";
}

sw.WriteLine("=====");
sw.WriteLine("ROADS finished data retrieval " + DateTime.Now);
sw.WriteLine("=====");
sw.Flush();
sw.Close();
txtLog.Text = txtLog.Text + ("=====") + "\n";
txtLog.Text = txtLog.Text + ("ROADS finished data retrieval " + DateTime.Now) + "\n";
txtLog.Text = txtLog.Text + ("=====") + "\n";
}
}
}

```

Appendix 10 – Query.cs

```

using System;
using System.Linq;
using System.Text;
using System.IO;
using System.Net;
using System.Data;
using System.Collections;

/// <summary>
/// Queries databases and puts all results into a SortedList
/// </summary>
public class Query {
    // CENTRALISED VARIABLES
    private SortedList resultsList;

    public Query() {
    }

    /// <summary>
    /// Collects query results into a list
    /// </summary>
    /// <returns>SortedList</returns>
    public SortedList queryAllSelectedDatabases(SortedList lstSelectedDatabases, StreamWriter sw) {

        try {
            sw.WriteLine("Querying for data...");
            sw.WriteLine("");
            // For each selected database (in selected database list) send query, retrieve response and collect into the SortedList
            resultsList = new SortedList();
            IEnumerator databasesEnum = lstSelectedDatabases.GetEnumerator();
            while (databasesEnum.MoveNext()) {
                DictionaryEntry dataSourceEntry = (DictionaryEntry)databasesEnum.Current;
                if (dataSourceEntry.Key.ToString() == "Mast") {
                    sw.WriteLine("Querying for Mast");
                    string resultsMastStr = getMastData(dataSourceEntry.Value.ToString());
                    resultsList.Add("Mast", resultsMastStr);
                }
                if (dataSourceEntry.Key.ToString() == "Simbad") {
                    sw.WriteLine("Querying for Simbad");
                    string resultsSimbadStr = getSimbadData(dataSourceEntry.Value.ToString());
                    resultsList.Add("Simbad", resultsSimbadStr);
                }
                if (dataSourceEntry.Key.ToString() == "OEC") {
                    sw.WriteLine("Querying for OEC");
                    string resultsOECStr = getOecData(dataSourceEntry.Value.ToString());
                    resultsList.Add("OEC", resultsOECStr);
                }
                if (dataSourceEntry.Key.ToString() == "OEC_XML") {
                    sw.WriteLine("Querying for OEC_XML");
                    string resultsOECStr = getOecXmlData(dataSourceEntry.Value.ToString());
                    resultsList.Add("OEC_XML", resultsOECStr);
                }
            }
            return resultsList;
        }
        catch (Exception ex) {
            sw.WriteLine("*** queryAllSelectedDatabases ERROR ***", DateTime.Now + " : ERROR during database querying: " + ex.Message);
            resultsList.Add("ERROR", DateTime.Now + " : ERROR during database querying: " + ex.Message);
            return resultsList;
        }
    }

    /// <summary>
    /// Requests data from website search
    /// </summary>
    /// <param name="queryStr"></param>
    /// <returns>The data in a string</returns>
    public string getOecData(string queryStr) {

```

```

string outputStr = "";
string urlAddress = queryStr;

//WebProxy webproxy = new WebProxy("http://161.112.232.103:3128");
HttpRequest request = WebRequest.Create(urlAddress) as HttpRequest;
//request.Proxy = webproxy;
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
Stream receiveStream = response.GetResponseStream();
Encoding encode = System.Text.Encoding.GetEncoding("utf-8");
// Pipes the stream to a higher level stream reader with the required encoding format.
StreamReader readStream = new StreamReader(receiveStream, encode);
Char[] read = new Char[256];
// Reads 256 characters at a time.
int count = readStream.Read(read, 0, 256);
while (count > 0) {
    // Adds the 256 characters onto a string.
    String str = new String(read, 0, count);
    outputStr = outputStr + str;
    count = readStream.Read(read, 0, 256);
}
// Releases the resources of the response.
response.Close();
// Releases the resources of the Stream.
readStream.Close();
// Configure string for XSLT use
string dateStr = DateTime.Today.ToShortDateString().Replace("/", " ");
//Trim faulty returned data
outputStr = outputStr.Substring(outputStr.IndexOf("<h3>Planet sizes</h3>"));
outputStr = outputStr.Substring(0, outputStr.IndexOf("<!-- ##### -->"));
outputStr = outputStr.Replace(""\"", "");
outputStr = outputStr.Replace("red" /><g style="stroke:black;">, "red" /><g style="stroke:black;">);

outputStr = "<root><device>OEC</device><location>GITHUB</location><target>Tau
Ceti</target><targetephemeris></targetephemeris><datetime>"
+ dateStr +
"</datetime><weather></weather><observer></observer><data>![CDATA[" + outputStr +
"]]</data><datatype>HTML</datatype><images></images><comments>For further info check the data</comments></root>";

return outputStr;
}

/// <summary>
/// Requests data from website search
/// </summary>
/// <param name="queryStr"></param>
/// <returns>The data in a string</returns>
public string getOecXmlData(string queryStr) {
    string outputStr = "";
    string urlAddress = queryStr;

    //WebProxy webproxy = new WebProxy("http://161.112.232.103:3128");
    HttpRequest request = WebRequest.Create(urlAddress) as HttpRequest;
    //request.Proxy = webproxy;
    HttpWebResponse response = (HttpWebResponse)request.GetResponse();
    Stream receiveStream = response.GetResponseStream();
    Encoding encode = System.Text.Encoding.GetEncoding("utf-8");
    // Pipes the stream to a higher level stream reader with the required encoding format.
    StreamReader readStream = new StreamReader(receiveStream, encode);
    Char[] read = new Char[256];
    // Reads 256 characters at a time.
    int count = readStream.Read(read, 0, 256);
    while (count > 0) {
        // Adds the 256 characters onto a string.
        String str = new String(read, 0, count);
        outputStr = outputStr + str;
        count = readStream.Read(read, 0, 256);
    }
    // Releases the resources of the response.
    response.Close();
    // Releases the resources of the Stream.
    readStream.Close();

```

```

    // Configure string for XSLT use
    string dateStr = DateTime.Today.ToShortDateString().Replace("/", " ");
    outputStr = "<root><device>OEC</device><location>GITHUB</location><target>Tau
Ceti</target><targetephemeris></targetephemeris><datetime>"
        + dateStr +
        "</datetime><weather></weather><observer></observer><data><![CDATA[" + outputStr +
        "]]></data><datatype>XML</datatype><images></images><comments>For further info check the data</comments></root>";

    return outputStr;
}

/// <summary>
/// Requests data from website search
/// </summary>
/// <param name="queryStr"></param>
/// <returns>The data in a string</returns>
public string getMastData(string queryStr) {
    string outputStr = "";
    string urlAddress = queryStr;

    //WebProxy webproxy = new WebProxy("http://161.112.232.103:3128");
    HttpRequest request = HttpRequest.Create(urlAddress) as HttpRequest;
    //request.Proxy = webproxy;
    HttpResponse response = (HttpResponse)request.GetResponse();
    Stream receiveStream = response.GetResponseStream();
    Encoding encode = System.Text.Encoding.GetEncoding("utf-8");
    // Pipes the stream to a higher level stream reader with the required encoding format.
    StreamReader readStream = new StreamReader(receiveStream, encode);
    Char[] read = new Char[256];
    // Reads 256 characters at a time.
    int count = readStream.Read(read, 0, 256);
    while (count > 0) {
        // Adds the 256 characters onto a string.
        String str = new String(read, 0, count);
        outputStr = outputStr + str;
        count = readStream.Read(read, 0, 256);
    }
    // Releases the resources of the response.
    response.Close();
    // Releases the resources of the Stream.
    readStream.Close();
    // Configure string for XSLT use
    string dateStr = DateTime.Today.ToShortDateString().Replace("/", " ");
    outputStr = "<root><device>MAST Repository</device><location>USA</location><target>Tau
Ceti</target><targetephemeris></targetephemeris><datetime>"
        + dateStr +
        "</datetime><weather></weather><observer></observer><data>" + outputStr +
        "</data><datatype>CSV</datatype><images></images><comments>For further info check the data</comments></root>";

    return outputStr;
}

/// <summary>
/// Requests data from website search
/// </summary>
/// <param name="queryStr"></param>
/// <returns>The data in a string</returns>
public string getSimbadData(string queryStr) {
    string outputStr = "";
    string urlAddress = queryStr;

    //WebProxy webproxy = new WebProxy("http://161.112.232.103:3128");
    HttpRequest request = HttpRequest.Create(urlAddress) as HttpRequest;
    //request.Proxy = webproxy;
    HttpResponse response = (HttpResponse)request.GetResponse();
    Stream receiveStream = response.GetResponseStream();
    Encoding encode = System.Text.Encoding.GetEncoding("utf-8");
    // Pipes the stream to a higher level stream reader with the required encoding format.
    StreamReader readStream = new StreamReader(receiveStream, encode);
    Char[] read = new Char[256];

```

```

// Reads 256 characters at a time.
int count = readStream.Read(read, 0, 256);
while (count > 0) {
    // Adds the 256 characters onto a string.
    String str = new String(read, 0, count);
    outputStr = outputStr + str;
    count = readStream.Read(read, 0, 256);
}
// Releases the resources of the response.
response.Close();
// Releases the resources of the Stream.
readStream.Close();
// First truncate SIMDAD result to only contain SIMBAD target star link data
outputStr = outputStr.Substring(outputStr.IndexOf("#notes"));
outputStr = outputStr.Substring(outputStr.IndexOf("http") - 1,
    (outputStr.IndexOf(">*")) - (outputStr.IndexOf("http") - 1));
// Configure string for XSLT use
string dateStr = DateTime.Today.ToShortDateString().Replace("/", " ");
outputStr = "<root><device>SIMBAD</device><location>Strasbourg</location><target>Tau
Ceti</target><targetephemeris></targetephemeris><datetime>"
    + dateStr +
    "</datetime><weather></weather><observer></observer><data><![CDATA[" + outputStr +
    "]]></data><datatype>SIMBADHTML</datatype><images></images><comments>For further info check the data</comments></root>";

return outputStr;
}
}

```

Appendix 11 – QueryTransform.cs

```

using System;
using System.Collections;
using System.Linq;
using System.Text;
using System.IO;
using System.Xml;
using System.Xml.XPath;
using System.Xml.Xsl;

/// <summary>
/// Carries out data transform from SortedList into an XmlDocument
/// </summary>
public class QueryTransform {
    // CENTRALISED VARIABLES
    XmlDocument resultsXmlDoc;
    string resultsXmlStr = "";
    IEnumerator resultsEnum;

    public QueryTransform() {
    }

    /// <summary>
    /// Transforms data from SortedList into a single XmlDocument
    /// </summary>
    /// <param name="queryResults"></param>
    /// <param name="sw"></param>
    /// <returns>resultsXmlDoc</returns>
    public XmlDocument transformIntoXmlDoc(SortedList queryResults, StreamWriter sw) {
        // Convert each datasource item and enter into xml document

        resultsEnum = queryResults.GetEnumerator();
        string resultStr = "";
        while (resultsEnum.MoveNext()) {
            DictionaryEntry deQuery = (DictionaryEntry)resultsEnum.Current;
            string keyStr = deQuery.Key.ToString();
            string valueStr = deQuery.Value.ToString();

            if (keyStr.Equals("Mast")) {
                // Transform MAST results and add to xml document string
                try {
                    // Create an XPathDocument using the XML string
                    StringReader stream = new StringReader(valueStr);
                    XPathDocument xpathDoc = new XPathDocument(stream);
                    // Create the new transform object
                    XslCompiledTransform transform = new XslCompiledTransform();

                    // Collect the resulting transformed XML in StringBuilder object
                    StringBuilder resultString = new StringBuilder();
                    XmlWriter writer = XmlWriter.Create(resultString);

                    transform.Load("C:\\ROAD\\RetrievalOfAstronomicalData\\RetrievalOfAstronomicalData\\RetrievalOfAstronomicalData\\mastQuery.xslt");
                    transform.Transform(xpathDoc, writer);

                    // String to store the resulting transformed XML
                    resultStr = resultString.ToString();
                    resultStr = resultStr.Replace("utf-16", "utf-8");

                    // Add to resultsXmlStr
                    if (resultsXmlStr.Equals("")) resultsXmlStr = resultStr;
                    else {
                        // Insert into existing text string
                        resultStr = resultStr.Substring(resultStr.IndexOf("<observationalastronomy>"),
                            (resultStr.IndexOf("</observationalastronomy>") + 25) - resultStr.IndexOf("<observationalastronomy>"));
                        resultsXmlStr = resultsXmlStr.Substring(0, resultsXmlStr.IndexOf("<observationalastronomy>")) +
                            resultStr +
                            resultsXmlStr.Substring(resultsXmlStr.IndexOf("<observationalastronomy>"));
                    }
                }
                catch (System.Exception ex) {
                    sw.WriteLine(DateTime.Now.ToString() + "**** ERROR On transformIntoXmlDoc 'Mast transform' *** Exception: ");
                }
            }
        }
    }
}

```

```

        + ex.Message + " *** "
        + ex.StackTrace);
    sw.Flush();
}
}
if (keyStr.Equals("Simbad")) {
    // Transform SIMBAD results and add to resultsXmlStr
    try {
        // Create an XPathDocument using the XML string
        StringReader stream = new StringReader(valueStr);
        XPathDocument xpathDoc = new XPathDocument(stream);
        // Create the new transform object
        XslCompiledTransform transform = new XslCompiledTransform();

        // Collect the resulting transformed XML in StringBuilder object
        StringBuilder resultString = new StringBuilder();
        XmlWriter writer = XmlWriter.Create(resultString);

transform.Load("C:\\ROAD\\RetrievalOfAstronomicalData\\RetrievalOfAstronomicalData\\RetrievalOfAstronomicalData\\simbadQuery.xslt");
transform.Transform(xpathDoc, writer);

        // String to store the resulting transformed XML
        resultStr = resultString.ToString();
        resultStr = resultStr.Replace("utf-16", "utf-8");

        // Add to resultsXmlStr
        if (resultsXmlStr.Equals("")) resultsXmlStr = resultStr;
        else {
            // Insert into existing text string
            resultStr = resultStr.Substring(resultStr.IndexOf("<observationalastronomy>"),
                (resultStr.IndexOf("</observationalastronomy>") + 25) - resultStr.IndexOf("<observationalastronomy>"));
            resultsXmlStr = resultsXmlStr.Substring(0, resultsXmlStr.IndexOf("<observationalastronomy>")) +
                resultStr +
                resultsXmlStr.Substring(resultsXmlStr.IndexOf("<observationalastronomy>"));
        }
    }
    catch (System.Exception ex) {
        sw.WriteLine(DateTime.Now.ToString() + "*** ERROR On transformIntoXmlDoc 'Simbad transform' *** Exception: "
            + ex.Message + " *** "
            + ex.StackTrace);

        sw.Flush();
    }
}
if (keyStr.Equals("OEC")) {
    // Transform OAC results and add to resultsXmlStr
    try {
        // Create an XPathDocument using the XML string
        StringReader stream = new StringReader(valueStr);
        XPathDocument xpathDoc = new XPathDocument(stream);
        // Create the new transform object
        XslCompiledTransform transform = new XslCompiledTransform();

        // Collect the resulting transformed XML in StringBuilder object
        StringBuilder resultString = new StringBuilder();
        XmlWriter writer = XmlWriter.Create(resultString);

transform.Load("C:\\ROAD\\RetrievalOfAstronomicalData\\RetrievalOfAstronomicalData\\RetrievalOfAstronomicalData\\oecQuery.xslt");
transform.Transform(xpathDoc, writer);

        // String to store the resulting transformed XML
        resultStr = resultString.ToString();
        resultStr = resultStr.Replace("utf-16", "utf-8");

        // Add to resultsXmlStr
        if (resultsXmlStr.Equals("")) resultsXmlStr = resultStr;
        else {
            // Insert into existing text string
            resultStr = resultStr.Substring(resultStr.IndexOf("<observationalastronomy>"),
                (resultStr.IndexOf("</observationalastronomy>") + 25) - resultStr.IndexOf("<observationalastronomy>"));
            resultsXmlStr = resultsXmlStr.Substring(0, resultsXmlStr.IndexOf("<observationalastronomy>")) +
                resultStr +
                resultsXmlStr.Substring(resultsXmlStr.IndexOf("<observationalastronomy>"));
        }
    }
    catch (System.Exception ex) {
        sw.WriteLine(DateTime.Now.ToString() + "*** ERROR On transformIntoXmlDoc 'OEC transform' *** Exception: "
            + ex.Message + " *** "
            + ex.StackTrace);

        sw.Flush();
    }
}
}
}

```



```

    }
}
catch (System.Exception ex) {
    sw.WriteLine(DateTime.Now.ToString() + "*** ERROR On transformIntoXmlDoc 'OEC transform' *** Exception: "
        + ex.Message + " *** "
        + ex.StackTrace);

    sw.Flush();
}
}
if (keyStr.Equals("OEC_XML")) {
    // Transform OAC_XML results and add to resultsXmlStr
    try {
        // Create an XPathDocument using the XML string
        StringReader stream = new StringReader(valueStr);
        XPathDocument xpathDoc = new XPathDocument(stream);
        // Create the new transform object
        XslCompiledTransform transform = new XslCompiledTransform();

        // Collect the resulting transformed XML in StringBuilder object
        StringBuilder resultString = new StringBuilder();
        XmlWriter writer = XmlWriter.Create(resultString);

transform.Load("C:\\ROAD\\RetrievalOfAstronomicalData\\RetrievalOfAstronomicalData\\RetrievalOfAstronomicalData\\oecQuery.xslt");
transform.Transform(xpathDoc, writer);

        // String to store the resulting transformed XML
        resultStr = resultString.ToString();
        resultStr = resultStr.Replace("utf-16", "utf-8");
        resultStr = resultStr.Replace("![CDATA[", "");
        resultStr = resultStr.Replace("]]", "");

        // Add to resultsXmlStr
        if (resultsXmlStr.Equals("")) resultsXmlStr = resultStr;
        else {
            // Insert into existing text string
            resultStr = resultStr.Substring(resultStr.IndexOf("<observationalastronomy>"),
                (resultStr.IndexOf("</observationalastronomy>") + 25) - resultStr.IndexOf("<observationalastronomy>"));
            resultsXmlStr = resultsXmlStr.Substring(0, resultsXmlStr.IndexOf("<observationalastronomy>")) +
                resultStr +
                resultsXmlStr.Substring(resultsXmlStr.IndexOf("<observationalastronomy>"));
        }
    }
    catch (System.Exception ex) {
        sw.WriteLine(DateTime.Now.ToString() + "*** ERROR On transformIntoXmlDoc 'OEC_XML transform' *** Exception: "
            + ex.Message + " *** "
            + ex.StackTrace);

        sw.Flush();
    }
}
}

// Create XmlDocument from resultsXmlStr
resultsXmlDoc = new XmlDocument();
resultsXmlDoc.LoadXml(resultsXmlStr);

return resultsXmlDoc;
}
}

```

Appendix 12 – OutputTransform.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.IO;
using System.Net;
using System.Data;
using System.Collections;
using System.Xml;
using System.Xml.Schema;
using System.Xml.XPath;
using System.Xml.Xsl;
using System.Xml.Linq;

public class OutputTransform {
    // CENTRALISED VARIABLES
    XmlDocument resultsXmlDoc;
    string resultsXmlStr = "";
    string htmlTransformStr = "";
    string pdfTransformStr = "";
    private StreamWriter sw2;

    public OutputTransform() {
    }

    /// <summary>
    /// Calls the required output processes
    /// </summary>
    /// <param name="chosenOutputs"></param>
    /// <param name="dataDoc"></param>
    /// <param name="sw"></param>
    public void createOutputs(SortedList chosenOutputs, XmlDocument dataDoc, StreamWriter sw) {
        try {
            sw.WriteLine("");
            sw.WriteLine("Getting output choices...");
            // For each selected output choice, do transforms and create output file
            IEnumerator outputsEnum = chosenOutputs.GetEnumerator();
            while (outputsEnum.MoveNext()) {
                DictionaryEntry outputEntry = (DictionaryEntry)outputsEnum.Current;
                if (outputEntry.Key.ToString() == "PDF") {
                    sw.WriteLine("");
                    sw.WriteLine("Choosing PDF");
                    outputPdf(dataDoc, sw);
                }
                if (outputEntry.Key.ToString() == "HTML") {
                    sw.WriteLine("");
                    sw.WriteLine("Choosing HTML");
                    outputHtml(dataDoc, sw);
                }
                if (outputEntry.Key.ToString() == "Machine") {
                    sw.WriteLine("");
                    sw.WriteLine("Choosing Machine");
                    outputMachine(dataDoc, sw);
                }
            }
        }
        catch (Exception ex) {
            sw.WriteLine("*** getChosenOutputs ERROR ***", DateTime.Now + " : ERROR during creation of output files: " + ex.Message);
        }
    }

    /// <summary>
    /// Creates HTML file using xmlDocument
    /// </summary>
    public void outputHtml(XmlDocument dataDoc, StreamWriter sw) {

```

```

// Get the contents of the <data> tags
// Assemble them into an HTML page using an XSLT file if necessary
// Save to file

// Transform
try {
    // Create an XPathDocument using the XML string
    string tempStr2 = dataDoc.OuterXml;
    tempStr2 = tempStr2.Replace("![CDATA[", "");
    tempStr2 = tempStr2.Replace("]]", "");
    tempStr2 = tempStr2.Replace("<name>tau Ceti b</name>", "");
    tempStr2 = tempStr2.Replace("<name>tau Ceti c</name>", "");
    tempStr2 = tempStr2.Replace("<name>tau Ceti d</name>", "");
    tempStr2 = tempStr2.Replace("<name>tau Ceti e</name>", "");
    tempStr2 = tempStr2.Replace("<name>tau Ceti f</name>", "");

    // Reset xml tags for OEC_XML
    string tempStr3 = tempStr2.Substring(tempStr2.IndexOf("&lt;system&gt;"),
                                         (tempStr2.IndexOf("&lt;/system&gt;") + 15) - tempStr2.IndexOf("&lt;system&gt;"));
    tempStr3 = tempStr3.Replace("&lt;", "<");
    tempStr3 = tempStr3.Replace("&gt;", ">");
    tempStr2 = tempStr2.Substring(0, tempStr2.IndexOf("&lt;system&gt;")) +
        tempStr3 +
        tempStr2.Substring(tempStr2.IndexOf("&lt;/system&gt;") + 15);

    StringReader stream = new StringReader(tempStr2);
    XPathDocument xpathDoc = new XPathDocument(stream);
    // Create the new transform object
    XslCompiledTransform transform = new XslCompiledTransform();

    // Collect the resulting transformed XML in StringBuilder object

    StringBuilder resultString = new StringBuilder();
    XmlWriter writer = XmlWriter.Create(resultString);

transform.Load("C:\\ROAD\\RetrievalOfAstronomicalData\\RetrievalOfAstronomicalData\\RetrievalOfAstronomicalData\\htmlOutput.xslt");
transform.Transform(xpathDoc, writer);

    // String to store the resulting transformed XML
    htmlTransformStr = resultString.ToString();
    htmlTransformStr = htmlTransformStr.Replace("&lt;", "<");
    htmlTransformStr = htmlTransformStr.Replace("&gt;", ">");
    htmlTransformStr = htmlTransformStr.Replace("%20", " ");
    htmlTransformStr = htmlTransformStr.Replace("%40", "@");
    htmlTransformStr = htmlTransformStr.Replace("utf-16", "utf-8");

    // Format Mast data from csv type to html table compliant
    if (htmlTransformStr.Contains("MAST")) {
        string tempStr = htmlTransformStr.Substring(htmlTransformStr.IndexOf("<Table>") + 7,
                                                    htmlTransformStr.IndexOf("</Table>") - (htmlTransformStr.IndexOf("<Table>") + 7));
        string[] tempStrArray = tempStr.Split('\n');
        IEnumerator tempEnum = tempStrArray.GetEnumerator();
        string changedStr = "";
        int counter = 0;
        while (tempEnum.MoveNext()) {
            if (counter != 1) changedStr = changedStr + "<tr>";
            string rowStr = (string)tempEnum.Current;
            rowStr = rowStr.Replace("\r", "");
            string[] rowStrArray = rowStr.Split(';');
            IEnumerator tempRowEnum = rowStrArray.GetEnumerator();
            while (tempRowEnum.MoveNext()) {
                if (counter == 0) {
                    changedStr = changedStr + "<th>" + (string)tempRowEnum.Current + "</th>";
                }
                if (counter > 1) {
                    changedStr = changedStr + "<td>" + (string)tempRowEnum.Current + "</td>";
                }
            }
            if (counter != 1) changedStr = changedStr + "</tr>";
            counter++;
        }
    }
}

```

```

        htmlTransformStr = htmlTransformStr.Substring(0, htmlTransformStr.IndexOf("<Table>") + 7) +
            changedStr +
            htmlTransformStr.Substring(htmlTransformStr.IndexOf("</Table>"));
    }
    // Save to file
    string timeStr = DateTime.Now.ToString();
    timeStr = timeStr.Replace("/", " ");
    timeStr = timeStr.Replace(":", "-");
    // Incorporate date time into file name
    string path = "C://roadLogs//outputs/htmlOutput_" + timeStr + ".html";
    // Create new folder if not exists.
    if (Directory.Exists("C://roadLogs//outputs") == false) {
        DirectoryInfo dir = Directory.CreateDirectory("C://roadLogs//outputs");
    }
    File.WriteAllText(@"C://roadLogs//outputs/htmlOutput_" + timeStr + ".html", htmlTransformStr);

    // Create a new stream to write to the text log file
    sw.WriteLine("htmlOutput created: " + "C://roadLogs//outputs/htmlOutput_" + timeStr + ".html");
    sw.WriteLine("");
}
catch (System.Exception ex) {
    sw.WriteLine(DateTime.Now.ToString() + "*** ERROR On OutputTransform.outputHtml 'htmlOutput transform' *** Exception: "
        + ex.Message + " *** "
        + ex.StackTrace);
    sw.Flush();
}
}

/// <summary>
/// Creates PDF file using xmlDocument
/// </summary>
public void outputPdf(XmlDocument dataDoc, StreamWriter sw) {
    // An XSL-FO transform is to be used here
    // Save to file

    // Transform
    try {
        // Create an XPathDocument using the XML string
        string tempStr2 = dataDoc.OuterXml;
        tempStr2 = tempStr2.Replace("![CDATA[", "");
        tempStr2 = tempStr2.Replace("]]", "");
        tempStr2 = tempStr2.Replace("<name>tau Ceti b</name>", "");
        tempStr2 = tempStr2.Replace("<name>tau Ceti c</name>", "");
        tempStr2 = tempStr2.Replace("<name>tau Ceti d</name>", "");
        tempStr2 = tempStr2.Replace("<name>tau Ceti e</name>", "");
        tempStr2 = tempStr2.Replace("<name>tau Ceti f</name>", "");

        // Reset xml tags for OEC_XML
        string tempStr3 = tempStr2.Substring(tempStr2.IndexOf("<system>"),
            (tempStr2.IndexOf("</system>") + 15) - tempStr2.IndexOf("<system>"));
        tempStr3 = tempStr3.Replace("<", "<");
        tempStr3 = tempStr3.Replace(">", ">");
        tempStr2 = tempStr2.Substring(0, tempStr2.IndexOf("<system>")) +
            tempStr3 +
            tempStr2.Substring(tempStr2.IndexOf("</system>") + 15);

        StringReader stream = new StringReader(tempStr2);
        XPathDocument xpathDoc = new XPathDocument(stream);
        // Create the new transform object
        XslCompiledTransform transform = new XslCompiledTransform();

        // Collect the resulting transformed XML in StringBuilder object
        StringBuilder resultString = new StringBuilder();
        XmlWriter writer = XmlWriter.Create(resultString);

        transform.Load("C://ROAD\\RetrievalOfAstronomicalData\\RetrievalOfAstronomicalData\\RetrievalOfAstronomicalData\\pdfOutput.xslt");
        transform.Transform(xpathDoc, writer);

        // String to store the resulting transformed XML
        pdfTransformStr = resultString.ToString();
        pdfTransformStr = pdfTransformStr.Replace("<", "<");
    }
}

```

```

pdfTransformStr = pdfTransformStr.Replace("&gt;", ">");
pdfTransformStr = pdfTransformStr.Replace("%20", " ");
pdfTransformStr = pdfTransformStr.Replace("%40", "@");
pdfTransformStr = pdfTransformStr.Replace("utf-16", "utf-8");
pdfTransformStr = pdfTransformStr.Replace("<svg", "<svg xmlns:svg=\"http://www.w3.org/2000/svg\"");

// Format Mast data from csv type to html table compliant
if (pdfTransformStr.Contains("MAST")) {
    string tempStr = pdfTransformStr.Substring(pdfTransformStr.IndexOf("Dataset,Target Name"),
        pdfTransformStr.IndexOf("</fo:table></fo:flow>") - (pdfTransformStr.IndexOf("Dataset,Target Name")));
    string[] tempStrArray = tempStr.Split('\n');
    IEnumerator tempEnum = tempStrArray.GetEnumerator();
    string changedStr = "";
    int counter = 0;
    int numberOfColumns = 0;
    while (tempEnum.MoveNext()) {
        if (counter != 1) changedStr = changedStr + "<fo:table-row>";
        string rowStr = (string)tempEnum.Current;
        rowStr = rowStr.Replace("\r", "");
        string[] rowStrArray = rowStr.Split(',');
        IEnumerator tempRowEnum = rowStrArray.GetEnumerator();
        while (tempRowEnum.MoveNext()) {
            if (counter == 0) {
                numberOfColumns++;
                changedStr = changedStr + "<fo:table-cell><fo:block color=\"blue\" font-weight=\"bold\">\" + (string)tempRowEnum.Current +
"</fo:block></fo:table-cell>";
            }
            if (counter > 1) {
                changedStr = changedStr + "<fo:table-cell><fo:block>\" + (string)tempRowEnum.Current + "</fo:block></fo:table-cell>";
            }
        }
        if (counter != 1) changedStr = changedStr + "</fo:table-row>";
        counter++;
    }
    // Create string for table column creation
    string columnCreationStr = "";
    for (int i = 1; i < numberOfColumns; i++) {
        columnCreationStr = columnCreationStr + "<fo:table-column column-width=\"15mm\"/>";
    }
    pdfTransformStr = pdfTransformStr.Substring(0, pdfTransformStr.IndexOf("Dataset,Target Name")) +
        columnCreationStr +
        "<fo:table-body>\" +
        changedStr +
        "</fo:table-body>\" +
        pdfTransformStr.Substring(pdfTransformStr.IndexOf("</fo:table></fo:flow>"));
}

// Declare svg fully for xsl-fo transform
if (pdfTransformStr.Contains("<svg>")) {
    pdfTransformStr = pdfTransformStr.Replace("<svg xmlns:svg=\"http://www.w3.org/2000/svg\" width=\"480\" height=\"400\">\",
"</fo:block><fo:block><fo:instream-foreign-object><svg:svg xmlns:svg=\"http://www.w3.org/2000/svg\" width=\"480\" height=\"400\">\";
"</fo:block><fo:instream-foreign-object><svg:svg xmlns:svg=\"http://www.w3.org/2000/svg\" width=\"600\" height=\"100\">\";
size=\"10pt\" font-family=\"sans-serif\" line-height=\"15pt\" space-after.optimum=\"3pt\" text-align=\"center\">\";
pdfTransformStr = pdfTransformStr.Replace("<g style=\"stroke:black\">\", "<svg:g style=\"stroke:black\">");
pdfTransformStr = pdfTransformStr.Replace("<g>\", "<svg:g>");
pdfTransformStr = pdfTransformStr.Replace("</g>\", "</svg:g>");
pdfTransformStr = pdfTransformStr.Replace("<text>\", "<svg:text>");
pdfTransformStr = pdfTransformStr.Replace("</text>\", "</svg:text>");
pdfTransformStr = pdfTransformStr.Replace("<circle>\", "<svg:circle>");
pdfTransformStr = pdfTransformStr.Replace("<line>\", "<svg:line>");
pdfTransformStr = pdfTransformStr.Replace("style=\"fill:url(#g2); stroke:none\", \"fill=\"lightgrey\"");
pdfTransformStr = pdfTransformStr.Replace("style=\"fill:url(#g3); stroke:none\", \"fill=\"brown\"");
pdfTransformStr = pdfTransformStr.Replace("ellipse\", \"svg:ellipse");
pdfTransformStr = pdfTransformStr.Replace("<svg xmlns:svg=\"http://www.w3.org/2000/svg\" width=\"600\" height=\"100\">\",
"</fo:block><fo:block><fo:instream-foreign-object><svg:svg xmlns:svg=\"http://www.w3.org/2000/svg\" width=\"600\" height=\"100\">\";
pdfTransformStr = pdfTransformStr.Replace("</fo:block></fo:flow></fo:page-sequence></fo:root>\", "</fo:block></fo:flow></fo:page-
sequence></fo:root>");
pdfTransformStr = pdfTransformStr.Replace("<p>\", \"");
pdfTransformStr = pdfTransformStr.Replace("</p>\", \"");
pdfTransformStr = pdfTransformStr.Replace("<h3>\", \"");
pdfTransformStr = pdfTransformStr.Replace("</h3>\", \"");
pdfTransformStr = pdfTransformStr.Replace("fill=\"url(#habitablegradient)\"\", \"fill=\"lightgreen\"");
pdfTransformStr = pdfTransformStr.Replace("<g style=\"stroke:black\">\", "<svg:g style=\"stroke:black\">");

```

```

    }
    // Save to file
    string timeStr = DateTime.Now.ToString();
    timeStr = timeStr.Replace("/", " ");
    timeStr = timeStr.Replace(":", "-");
    // Incorporate date time into file name
    string path = "C://roadLogs//outputs//pdfOutput_" + timeStr + ".fo";
    // Create new folder if not exists.
    if (Directory.Exists("C://roadLogs//outputs") == false) {
        DirectoryInfo dir = Directory.CreateDirectory("C://roadLogs//outputs");
    }
    File.WriteAllText(@"C://roadLogs//outputs//pdfOutput_" + timeStr + ".fo", pdfTransformStr);

    // Create a new stream to write to the text log file
    sw.WriteLine("pdfOutput created: " + "C://roadLogs//outputs//pdfOutput_" + timeStr + ".fo");
    sw.WriteLine("");
}
catch (System.Exception ex) {
    sw.WriteLine(DateTime.Now.ToString() + "*** ERROR On OutputTransform.outputPdf 'pdfOutput transform' *** Exception: "
        + ex.Message + " *** "
        + ex.StackTrace);
    sw.Flush();
}
}

/// <summary>
/// Creates file suitable for machine transfer using xmlDocument
/// </summary>
public void outputMachine(XmlDocument dataDoc, StreamWriter sw) {
    // Just pass the XmlDocument through without alteration... as is machine readable
    // Save straight to file
    try {
        // Get XmlDocument string and write to file
        string tempStr2 = dataDoc.OuterXml;

        // Save to file
        string timeStr = DateTime.Now.ToString();
        timeStr = timeStr.Replace("/", " ");
        timeStr = timeStr.Replace(":", "-");
        // Incorporate date time into file name
        string path = "C://roadLogs//outputs//machineOutput_" + timeStr + ".xml";
        // Create new folder if not exists.
        if (Directory.Exists("C://roadLogs//outputs") == false) {
            DirectoryInfo dir = Directory.CreateDirectory("C://roadLogs//outputs");
        }
        File.WriteAllText(@"C://roadLogs//outputs//machineOutput_" + timeStr + ".xml", tempStr2);

        // Create a new stream to write to the text log file
        //sw2 = File.AppendText(path);
        //sw2.Write(htmlTransformStr);
        sw.WriteLine("machineOutput created: " + "C://roadLogs//outputs//machineOutput_" + timeStr + ".xml");
        sw.WriteLine("");
    }
    catch (System.Exception ex) {
        sw.WriteLine(DateTime.Now.ToString() + "*** ERROR On OutputTransform.outputMachine 'machineOutput transform' *** Exception: "
            + ex.Message + " *** "
            + ex.StackTrace);
        sw.Flush();
    }
}
}
}

```

Appendix 13 – mastQuery.xslt

```

<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"

xmlns:roNamespaceSchemaLocation="file://C://ROAD//RetrievalOfAstronomicalData//RetrievalOfAstronomicalData//astronomy.xsd">
  <xsl:template match="/">
    <astronomy>
      <astrophysics></astrophysics>
      <astrobiology></astrobiology>
      <astrochemistry></astrochemistry>
      <archeoastronomy></archeoastronomy>
      <planetaryscience></planetaryscience>
      <observationalastronomy>
        <infrared>
          <device>
            <manufacturer>

            </manufacturer>
            <model>
              <xsl:value-of select="root/device"/>
            </model>
          </device>
          <location>
            <gridRef>

            </gridRef>
            <address>
              <xsl:value-of select="root/location"/>
            </address>
          </location>
          <target>
            <targetname>
              <xsl:value-of select="root/target"/>
            </targetname>
            <targetposition>

            </targetposition>
          </target>
          <targetephemeris>
            <ephemeris>
              <xsl:value-of select="root/targetephemeris"/>
            </ephemeris>
          </targetephemeris>
          <datetime>
            <thedata>
              <xsl:value-of select="root/datetime"/>
            </thedata>
            <thetime>

            </thetime>
          </datetime>
          <weather>
            <weatherdesc>
              <xsl:value-of select="root/weather"/>
            </weatherdesc>
          </weather>
          <observer>
            <forename>

            </forename>
            <surname>
              <xsl:value-of select="root/observer"/>
            </surname>
          </observer>
          <irobservationdata>
            <thetime></thetime>
            <theintensity></theintensity>
            <thewavelength></thewavelength>
          </irobservationdata>
        </infrared>
      </observationalastronomy>
    </astronomy>
  </template>

```

```

<description>
  <xsl:value-of select="root/datatype"/>
</description>
<fileurlir></fileurlir>
<filenameeir></filenameeir>
<thedatair>
  <xsl:value-of select="root/data"/>
</thedatair>
<thedatairblob></thedatairblob> <!-- blob type Base 64 encoding -->
</irobservationdata>
<images>
<fileurl>
  <xsl:value-of select="root/images"/>
</fileurl>
<filename>

</filename>
<imagedata>

</imagedata>
</images>
<comments>
<thecomments>
  <xsl:value-of select="root/comments"/>
</thecomments>
</comments>
</infrared>
</observationalastronomy>
</astronomy>
</xsl:template>
</xsl:stylesheet>

```


Appendix 14 – oecQuery.xslt

```

<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"

xmlns:noNamespaceSchemaLocation="file://C://ROAD//RetrievalOfAstronomicalData//RetrievalOfAstronomicalData//astronomy.xsd">
<xsl:template match="/">
  <astronomy>
    <astrophysics></astrophysics>
    <astrobiology></astrobiology>
    <astrochemistry></astrochemistry>
    <archeoastronomy></archeoastronomy>
    <planetaryscience></planetaryscience>
    <observationalastronomy>
      <visible>
        <device>
          <manufacturer>

          </manufacturer>
          <model>
            <xsl:value-of select="root/device"/>
          </model>
        </device>
        <location>
          <gridRef>

          </gridRef>
          <address>
            <xsl:value-of select="root/location"/>
          </address>
        </location>
        <target>
          <targetname>
            <xsl:value-of select="root/target"/>
          </targetname>
          <targetposition>

          </targetposition>
        </target>
        <targetephemeris>
          <ephemeris>
            <xsl:value-of select="root/targetephemeris"/>
          </ephemeris>
        </targetephemeris>
        <datetime>
          <thedata>
            <xsl:value-of select="root/datetime"/>
          </thedata>
          <thetime>

          </thetime>
        </datetime>
        <weather>
          <weatherdesc>
            <xsl:value-of select="root/weather"/>
          </weatherdesc>
        </weather>
        <observer>
          <forename>

          </forename>
          <surname>
            <xsl:value-of select="root/observer"/>
          </surname>
          <contact>

          </contact>
        </observer>
        <visibleobservationdata>
          <iscolour></iscolour>
          <magnitude></magnitude>
          <description>

```

```

    <xsl:value-of select="root/datatype"/>
  </description>
  <fileurlvisible></fileurlvisible>
  <filenamevisible></filenamevisible>
  <thedatavisible>
    ![CDATA[<xsl:value-of select="root/data"/>]]
  </thedatavisible> <!-- USE CDATA -->
  <thedatavisibleblob></thedatavisibleblob> <!-- blob type Base 64 encoding -->
</visibleobservationdata>
<images>
  <fileurl>
    <xsl:value-of select="root/images"/>
  </fileurl>
  <filename>

  </filename>
  <imagedata>

  </imagedata>
</images>
<comments>
  <thecomments>
    <xsl:value-of select="root/comments"/>
  </thecomments>
</comments>
</visible>
</observationalastronomy>
</astronomy>
</xsl:template>
</xsl:stylesheet>

```

Appendix 15 – simbadQuery.xslt

```

<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"

xmlns:noNamespaceSchemaLocation="file://C://ROAD//RetrievalOfAstronomicalData//RetrievalOfAstronomicalData//astronomy.xsd">
<xsl:template match="/">
  <astronomy>
    <astrophysics></astrophysics>
    <astrobiology></astrobiology>
    <astrochemistry></astrochemistry>
    <archeoastronomy></archeoastronomy>
    <planetaryscience></planetaryscience>
    <observationalastronomy>
      <visible>
        <device>
          <manufacturer>

          </manufacturer>
          <model>
            <xsl:value-of select="root/device"/>
          </model>
        </device>
        <location>
          <gridRef>

          </gridRef>
          <address>
            <xsl:value-of select="root/location"/>
          </address>
        </location>
        <target>
          <targetname>
            <xsl:value-of select="root/target"/>
          </targetname>
          <targetposition>

          </targetposition>
        </target>
        <targetephemeris>
          <ephemeris>
            <xsl:value-of select="root/targetephemeris"/>
          </ephemeris>
        </targetephemeris>
        <datetime>
          <thedata>
            <xsl:value-of select="root/datetime"/>
          </thedata>
          <thetime>

          </thetime>
        </datetime>
        <weather>
          <weatherdesc>
            <xsl:value-of select="root/weather"/>
          </weatherdesc>
        </weather>
        <observer>
          <forename>

          </forename>
          <surname>
            <xsl:value-of select="root/observer"/>
          </surname>
          <contact>

          </contact>
        </observer>
        <visibleobservationdata>
          <iscolour></iscolour>
          <magnitude></magnitude>
          <description>

```

```

    <xsl:value-of select="root/datatype"/>
  </description>
  <fileurlvisible></fileurlvisible>
  <filenamevisible></filenamevisible>
  <thedatavisible>
    ![CDATA[<xsl:value-of select="root/data"/>]]
  </thedatavisible> <!-- USE CDATA -->
  <thedatavisibleblob></thedatavisibleblob> <!-- blob type Base 64 encoding -->
</visibleobservationdata>
<images>
  <fileurl>
    <xsl:value-of select="root/images"/>
  </fileurl>
  <filename>

  </filename>
  <imagedata>

  </imagedata>
</images>
<comments>
  <thecomments>
    <xsl:value-of select="root/comments"/>
  </thecomments>
</comments>
</visible>
</observationalastronomy>
</astronomy>
</xsl:template>
</xsl:stylesheet>

```

Appendix 16 – htmlOutput.xslt

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"

xmlns:noNamespaceSchemaLocation="file:///C:/ROAD/RetrievalOfAstronomicalData/RetrievalOfAstronomicalData/astronomy.xsd">

<xsl:template match="/">
<html>
<head>
<title>
<xsl:choose>
<xsl:when test="astronomy/observationalastronomy/infrared/target/targetname = "">
<xsl:value-of select="astronomy/observationalastronomy/visible/target/targetname"/>
</xsl:when>
<xsl:otherwise>
<xsl:value-of select="astronomy/observationalastronomy/infrared/target/targetname"/>
</xsl:otherwise>
</xsl:choose>
</title>
</head>
<body>
<table width="100%" style="height: 100%; cellpadding="10" cellspacing="0" border="0">
<tr>

<!-- ===== HEADER SECTION ===== -->
<td colspan="2" style="height: 80px;">
<h1>
<xsl:choose>
<xsl:when test="astronomy/observationalastronomy/infrared/target/targetname = "">
<xsl:value-of select="astronomy/observationalastronomy/visible/target/targetname"/>
</xsl:when>
<xsl:otherwise>
<xsl:value-of select="astronomy/observationalastronomy/infrared/target/targetname"/>
</xsl:otherwise>
</xsl:choose>
</h1>
</td>
</tr>

<xsl:apply-templates select="astronomy/observationalastronomy/visible"/>

<tr>
<td colspan="2" align="center" height="200">
<xsl:apply-templates select="astronomy/observationalastronomy/infrared"/>
</td>
</tr>

<!-- ===== FOOTER SECTION ===== -->
<tr>
<td colspan="2" align="center" height="20">
ROADS © Guy Beech 2015
</td>
</tr>

</table>
</body>
</html>
</xsl:template>

<xsl:template match="astronomy/observationalastronomy/visible">

<xsl:choose>
<xsl:when test="visibleobservationdata/description = 'HTML'">
<tr>
<td colspan="2" align="center" height="200">
<b>
<u>
<xsl:value-of select="device/model"/>
</u>
</b>
(Open Exoplanet Catalogue)
```

```

<p>
  <!-- NOTE: The SVG data in this data string is courtesy of Open Exoplanet Catalogue on GitHub-->
  <xsl:value-of select="visibleobservationdata/thedatavisible"/>
</p>
</td>
</tr>
</xsl:when>
<xsl:when test="visibleobservationdata/description = 'XML'">
  <tr>
    <td colspan="2" align="center" height="400">
      <b>
        <u>
          <xsl:value-of select="device/model"/>_XML
        </u>
      </b>
      (Open Exoplanet Catalogue - XML data)
    </td>

    <!-- HTML LAYOUT AND DATA IN HERE - DON'T NEED TO INITIALLY PUT THE CDATA TAGS ON FOR THIS ONE
    NO STAR DATA REQUIRED, JUST SYSTEM AND PLANET
    -->
    <p>
      Name: <b><xsl:value-of select="visibleobservationdata/thedatavisible/system/name"/></b>
      Right Ascension: <xsl:value-of select="visibleobservationdata/thedatavisible/system/rightascension"/>
      Declination: <xsl:value-of select="visibleobservationdata/thedatavisible/system/declination"/>
      Distance: <xsl:value-of select="visibleobservationdata/thedatavisible/system/distance"/>
    </p>
    <p>
      <b>
        Planets of <b><xsl:value-of select="visibleobservationdata/thedatavisible/system/name"/></b>
      </b>
    </p>
    <table>
      <tr>
        <th>Name</th>
        <th>List</th>
        <th>Mass</th>
        <th>Period</th>
        <th>Semi Major Axis</th>
        <th>Eccentricity</th>
        <th>Discovery Method</th>
        <th>Last Update</th>
        <th>Discovery Year</th>
        <th>Description</th>
      </tr>
      <xsl:for-each select="visibleobservationdata/thedatavisible/system/star/planet">
        <tr>
          <td>
            <xsl:value-of select="name"/>
          </td>
          <td>
            <xsl:value-of select="list"/>
          </td>
          <td>
            <xsl:value-of select="mass"/>
          </td>
          <td>
            <xsl:value-of select="period"/>
          </td>
          <td>
            <xsl:value-of select="semimajoraxis"/>
          </td>
          <td>
            <xsl:value-of select="eccentricity"/>
          </td>
          <td>
            <xsl:value-of select="discoverymethod"/>
          </td>
          <td>
            <xsl:value-of select="lastupdate"/>
          </td>
          <td>
            <xsl:value-of select="discoveryyear"/>

```

```

        </td>
        <td>
          <xsl:value-of select="description"/>
        </td>
      </tr>
    </xsl:for-each>
  </table>
  <p>
  </p>
  </td>
</tr>
</xsl:when>
<xsl:when test="visibleobservationdata/description = 'SIMBADHTML'">
  <tr>
    <td colspan="2" align="center" height="200">
      <b>
        <u>
          <xsl:value-of select="device/model"/>
        </u>
      </b>
      (Strasbourg astronomical Data Center)
    </td>
    <td>
      Take this link for SIMBAD star data and visual image provided by Aladin Lite (sky atlas)

      <!-- ONLY TAKING THE ALADIN LITE LINK FOR THE TARGET... all other data discarded -->
      &lt;A HREF=<xsl:value-of select="visibleobservationdata/thedatavisible"/>&gt;Tau Ceti&lt;/A&gt;

    </td>
  </tr>
</xsl:when>
</xsl:choose>
</xsl:template>

<xsl:template match="astronomy/observationalastronomy/infrared">
  <b>
    <u>
      From the <xsl:value-of select="device/model"/>
    </u>
  </b>
  (IR data - csv)
  <p>
    <Table>
      <xsl:value-of select="irobservationdata/thedatair"/>
    </Table>
  </p>
</xsl:template>
</xsl:stylesheet>

```

Appendix 17 – pdfOutput.xslt

```

<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:fo="http://www.w3.org/1999/XSL/Format"
    xmlns:svg="http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd"

xmlns:noNamespaceSchemaLocation="file://C://ROAD//RetrievalOfAstronomicalData//RetrievalOfAstronomicalData//astronomy.xsd">

<!-- The root template -->
<xsl:template match="/">
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
<fo:layout-master-set>

<!-- layout information -->
<fo:simple-page-master master-name="simple"
    page-height="29.7cm"
    page-width="30cm"
    margin-top="1cm"
    margin-bottom="1cm"
    margin-left="1.0cm"
    margin-right="1.0cm">
<fo:region-body margin-top="1cm"/>
<fo:region-before extent="1cm"/>
<fo:region-after extent="1.5cm"/>
</fo:simple-page-master>
</fo:layout-master-set>
<!-- end: defines page layout -->

<fo:page-sequence master-reference="simple">

<fo:static-content flow-name="xsl-region-before">
<fo:block text-align="end"
    font-size="10pt"
    font-family="serif"
    line-height="14pt">
ROADS - p. <fo:page-number/>
</fo:block>
</fo:static-content>

<fo:static-content flow-name="xsl-region-after">
<fo:block text-align="center"
    font-size="10pt"
    font-family="serif"
    line-height="14pt">
ROAD © Guy Beech 2015
</fo:block>
</fo:static-content>

<fo:flow flow-name="xsl-region-body">

<xsl:choose>
<xsl:when test="astronomy/observationalastronomy/infrared/target/targetname = """>
<!-- this defines a title -->
<fo:block font-size="16pt"
    font-family="sans-serif"
    line-height="24pt"
    space-after.optimum="15pt"
    background-color="white"
    color="blue"
    text-align="left"
    padding-top="3pt">
<!--Tau Ceti-->
<xsl:value-of select="astronomy/observationalastronomy/visible/target/targetname"/>
</fo:block>
</xsl:when>
<xsl:otherwise>
<!-- this defines a title -->
<fo:block font-size="16pt"
    font-family="sans-serif"
    line-height="24pt"

```



```

        space-after.optimum="15pt"
        background-color="white"
        color="blue"
        text-align="left"
        padding-top="3pt">
<!--Tau Ceti-->
<xsl:value-of select="astronomy/observationalastronomy/infrared/target/targetname"/>
</fo:block>
</xsl:otherwise>
</xsl:choose>
<!-- Apply the visible and infrared templates -->
<xsl:apply-templates select="astronomy/observationalastronomy/visible"/>
<xsl:apply-templates select="astronomy/observationalastronomy/infrared"/>

</fo:flow>
<!-- closes the flow region-after element-->
</fo:page-sequence>
<!-- closes the page-sequence -->

</fo:root>
</xsl:template>

<!-- The visible template -->
<xsl:template match="astronomy/observationalastronomy/visible">

<xsl:choose>

<xsl:when test="visibleobservationdata/description = 'HTML'">
<!-- this defines paragraph text -->
<fo:block font-size="12pt"
        font-family="sans-serif"
        line-height="15pt"
        space-after.optimum="3pt"
        text-align="center">
<fo:inline text-decoration="underline">
<xsl:value-of select="device/model"/> (Open Exoplanet Catalogue)
</fo:inline>
</fo:block>
<!-- this defines paragraph text -->
<fo:block font-size="10pt" font-family="sans-serif" line-height="15pt" space-after.optimum="3pt" text-align="center">
<!-- NOTE: The SVG data in this data string is courtesy of Open Exoplanet Catalogue on GitHub-->
<xsl:value-of select="visibleobservationdata/thedatavisible"/>
</fo:block>
</xsl:when>

<xsl:when test="visibleobservationdata/description = 'XML'">
<!-- this defines paragraph text -->
<fo:block font-size="12pt"
        font-family="sans-serif"
        line-height="15pt"
        space-after.optimum="3pt"
        text-align="center">
<fo:inline text-decoration="underline">
<xsl:value-of select="device/model"/> _XML (Open Exoplanet Catalogue - XML data)
</fo:inline>
</fo:block>
<!-- HTML LAYOUT AND DATA IN HERE - DON'T NEED TO INITIALLY PUT THE CDATA TAGS ON FOR THIS ONE
NO STAR DATA REQUIRED, JUST SYSTEM AND PLANET
-->
<!-- this defines paragraph text -->
<fo:block font-size="10pt"
        font-family="sans-serif"
        line-height="15pt"
        space-after.optimum="3pt"
        text-align="center">
Right Ascension: <xsl:value-of select="visibleobservationdata/thedatavisible/system/rightascension"/>
Declination: <xsl:value-of select="visibleobservationdata/thedatavisible/system/declination"/>
Distance: <xsl:value-of select="visibleobservationdata/thedatavisible/system/distance"/>
</fo:block>
<!-- this defines paragraph text -->
<fo:block font-size="10pt"
        font-family="sans-serif"

```

```

        line-height="15pt"
        space-after.optimum="3pt"
        text-align="center">
Planets of <xsl:value-of select="visibleobservationdata/thedatavisible/system/name"/>
</fo:block>
<!-- table start -->
<!-- <fo:table border-width="0.5mm" border-style="solid" table-layout="fixed" width="100%" border-collapse="separate" text-
align="center"> -->
<fo:table text-align="center" font-size="8pt" font-family="Cordia New">
  <fo:table-column column-width="20mm"/>
  <fo:table-column column-width="20mm"/>
  <fo:table-column column-width="20mm"/>
  <fo:table-column column-width="20mm"/>
  <fo:table-column column-width="20mm"/>
  <fo:table-column column-width="20mm"/>
  <fo:table-column column-width="20mm"/>
  <fo:table-column column-width="20mm"/>
  <fo:table-column column-width="50mm"/>
<fo:table-body>
  <!-- table header -->
  <fo:table-row>
    <fo:table-cell>
      <fo:block color="blue" font-weight="bold">Name</fo:block>
    </fo:table-cell>
    <fo:table-cell>
      <fo:block color="blue" font-weight="bold">List</fo:block>
    </fo:table-cell>
    <fo:table-cell>
      <fo:block color="blue" font-weight="bold">Mass</fo:block>
    </fo:table-cell>
    <fo:table-cell>
      <fo:block color="blue" font-weight="bold">Period</fo:block>
    </fo:table-cell>
    <fo:table-cell>
      <fo:block color="blue" font-weight="bold">Semi Major Axis</fo:block>
    </fo:table-cell>
    <fo:table-cell>
      <fo:block color="blue" font-weight="bold">Eccentricity</fo:block>
    </fo:table-cell>
    <fo:table-cell>
      <fo:block color="blue" font-weight="bold">Discovery Method</fo:block>
    </fo:table-cell>
    <fo:table-cell>
      <fo:block color="blue" font-weight="bold">Last Update</fo:block>
    </fo:table-cell>
    <fo:table-cell>
      <fo:block color="blue" font-weight="bold">Discovery Year</fo:block>
    </fo:table-cell>
    <fo:table-cell>
      <fo:block color="blue" font-weight="bold">Description</fo:block>
    </fo:table-cell>
  </fo:table-row>
  <!-- table rows -->
  <xsl:for-each select="visibleobservationdata/thedatavisible/system/star/planet">
    <fo:table-row>
      <fo:table-cell >
        <fo:block>
          <xsl:value-of select="name"/>
        </fo:block>
      </fo:table-cell>
      <fo:table-cell >
        <fo:block>
          <xsl:value-of select="list"/>
        </fo:block>
      </fo:table-cell>
      <fo:table-cell >
        <fo:block>
          <xsl:value-of select="mass"/>
        </fo:block>
      </fo:table-cell>
      <fo:table-cell >

```

```

        <fo:block>
          <xsl:value-of select="period"/>
        </fo:block>
      </fo:table-cell>
    </fo:table-cell>
    <fo:block>
      <xsl:value-of select="semimajoraxis"/>
    </fo:block>
  </fo:table-cell>
  <fo:table-cell>
    <fo:block>
      <xsl:value-of select="eccentricity"/>
    </fo:block>
  </fo:table-cell>
  <fo:table-cell>
    <fo:block>
      <xsl:value-of select="discoverymethod"/>
    </fo:block>
  </fo:table-cell>
  <fo:table-cell>
    <fo:block>
      <xsl:value-of select="lastupdate"/>
    </fo:block>
  </fo:table-cell>
  <fo:table-cell>
    <fo:block>
      <xsl:value-of select="discoveryyear"/>
    </fo:block>
  </fo:table-cell>
  <fo:table-cell>
    <fo:block>
      <xsl:value-of select="description"/>
    </fo:block>
  </fo:table-cell>
</fo:table-row>
</xsl:for-each>
</fo:table-body>
</fo:table>
<!-- table end -->
</xsl:when>

<xsl:when test="visibleobservationdata/description = 'SIMBADHTML'">
  <!-- this defines paragraph text -->
  <fo:block font-size="12pt"
    font-family="sans-serif"
    line-height="15pt"
    space-after.optimum="3pt"
    text-align="center">
    <fo:inline text-decoration="underline">
      <xsl:value-of select="device/model"/> (Strasbourg astronomical Data Center)
    </fo:inline>
  </fo:block>

  <!-- this defines paragraph text -->
  <fo:block font-size="8pt"
    font-family="sans-serif"
    line-height="15pt"
    space-after.optimum="3pt"
    text-align="center">
    Take this link belowfor SIMBAD Tau Ceti star data and visual image provided by Aladin Lite (sky atlas)
  </fo:block>

  <!-- this defines paragraph text -->
  <fo:block id="reference"
    color="blue"
    font-size="8pt"
    font-family="sans-serif"
    line-height="15pt"
    space-after.optimum="15pt"
    text-align="center">
    <!-- ONLY TAKING THE ALADIN LITE LINK FOR THE TARGET... all other data discarded -->
    <fo:inline text-decoration="underline">

```

```

        <fo:basic-link>
          <xsl:attribute name="external-destination">
            <xsl:value-of select="visibleobservationdata/thedatavisible"/>
          </xsl:attribute>
          <xsl:apply-templates/>
        </fo:basic-link>
      </fo:inline>
    </fo:block>
  </xsl:when>

</xsl:choose>
</xsl:template>

<!-- The infrared template -->
<xsl:template match="astronomy/observationalastronomy/infrared">
  <!-- this defines paragraph text -->
  <fo:block font-size="12pt"
    font-family="sans-serif"
    line-height="15pt"
    space-after.optimum="3pt"
    text-align="center">
    <fo:inline text-decoration="underline">
      From the <xsl:value-of select="device/model"/> (IR data - csv)
    </fo:inline>
  </fo:block>

  <!-- Table start -->
  <!--<fo:table border-width="0.5mm" border-style="solid" table-layout="fixed" width="100%" border-collapse="separate" text-align="center">-
->
    <fo:table text-align="center" font-size="8pt" font-family="Cordia New">
      <xsl:value-of select="iobservationdata/thedatair"/>
    </fo:table>
  <!-- table end -->
</xsl:template>

</xsl:stylesheet>

```

Appendix 18 – Public databases

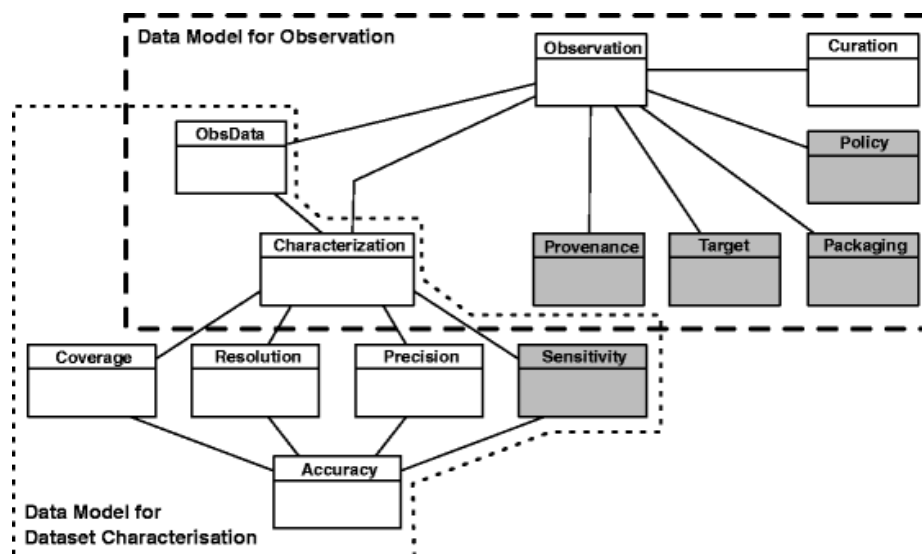
[Mikulski Archive for Space Telescopes \(MAST\)](#) (Mikulski Archive for Space Telescopes (MAST), 2015)
[Strasbourg astronomical Data Center - SIMBAD](#) (Strasbourg astronomical Data Center – SIMBAD, 2015)
[Open Exoplanet Catalogue \(OEC\)](#) (Open Exoplanet Catalogue, 2015)

The MAST archive is a repository of data archives from space based missions in the Ultraviolet/Optical/Near-IR range, based in Baltimore USA. MAST returned data in CSV format.

The SIMBAD astronomical database provides basic data, cross-identifications, bibliography and measurements for astronomical objects outside the solar system. It is supported and maintained by the Strasbourg Astronomical Data Center which hosts a number of catalogue databases. SIMBAD returned data in HTML format.

The Open Exoplanet Catalogue is a catalogue of all discovered extra-solar planets hosted on GitHub. It is decentralized and completely open. Contributions and corrections are welcomed from both professional astronomers and the general public. OEC returned data in XML format and HTML format.

Appendix 19 - The complete RADAMS data model



Appendix 20 - T.6. Table of Design Patterns

Design Pattern	Characteristics	Advantages	Disadvantages
Russian Doll	Contains only one global element. All other elements are local.	<ul style="list-style-type: none"> • Contains only one valid root element. • • Could reduce the complexity of namespace, depending on the elementFormDefault attribute of the schema. 	<ul style="list-style-type: none"> • Allows reuse for all or no elements. • • Supports single-file schemas only.
Salami Slice	Contains all global elements, hence many potential root elements.	<ul style="list-style-type: none"> • Contains all reusable elements. • • Supports reuse of elements from other documents. 	<ul style="list-style-type: none"> • Exposes the complexity in namespace. • • Renders the root difficult to determine.
Venetian Blind	Is an extension of Russian Doll and contains only one global element. All other elements are local.	<ul style="list-style-type: none"> • Contains only one single root element. • • Allows reuse for all the types and the single global element. • Allows multiple files. 	<ul style="list-style-type: none"> • Limits encapsulation by exposing types.
Garden of Eden	Is a combination of Venetian Blind and Salami Slice. All elements and types are global, hence many potential root elements.	<ul style="list-style-type: none"> • Allows reuse of both elements and types. • • Allows multiple files. 	<ul style="list-style-type: none"> • Contains many potential root elements. • • Limits encapsulation by exposing types. • • Is difficult to read and understand.

Appendix 21 - Table of ROAD Run Actions

Run 1	The maximum of all four data queries transformed and then output as all three possible types. All the output types to be saved as files, the schema validated XMLDocument saved as a file and the log file saved also. The XSL-FO file then separately rendered into a PDF and saved.
Run 2	MAST, SIMBAD, OEC_XML data queries transformed and then output as HTML. The output type to be saved as a file, the schema validated XMLDocument saved as a file and the log file saved also.
Run 3	SIMBAD, OEC_XML, OEC data queries transformed and then output as PDF. The output type to be saved as a file, the schema validated XMLDocument saved as a file and the log file saved also. The XSL-FO file then separately rendered into a PDF and saved.
Run 4	MAST, OEC_XML data queries transformed and then output as MACHINE. The output type to be saved as a file, the schema validated XMLDocument saved as a file and the log file saved also.
Run 5	<p>This run is designed to demonstrate the validation of the constructed XMLDocument by instigating a validation failure. An XSLT file is changed so as to produce an invalid structure in the XMLDocument. The resulting invalid status of the XMLDocument will then prevent any output files from being created.</p> <p>The maximum of all four data queries transformed and then output as all three possible types. All the output types to be saved as files, the schema validated XMLDocument saved as a file and the log file saved also. The XSL-FO file then separately rendered into a PDF and saved.</p>

10 References

- Open Exoplanet Catalogue. (2015, February 10). *Open Exoplanet Catalogue*. Retrieved from GitHub: <http://www.openexoplanetcatalogue.com/>
- Observational astronomy*. (2015, July 22). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Observational_astronomy
- Outline of astronomy*. (2015, July 22). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Outline_of_astronomy
- Andersen, R. (2014, November 14). *How Big Data Is Changing Astronomy (Again)*. Retrieved from The Atlantic Magazine: <http://www.theatlantic.com/technology/archive/2012/04/how-big-data-is-changing-astronomy-again/255917/>
- Annual Review of Astronomy and Astrophysics . (2014, December 10). *Annual Review of Astronomy and Astrophysics* . Retrieved from Annual Reviews: Journals: <http://www.annualreviews.org/journal/astro>
- Astronomy and Astrophysics. (2014, December 10). *Worldwide astronomical and astrophysical research*. Retrieved from Astronomy and Astrophysics: <http://www.aanda.org/>
- Astronomy Letters. (2014, December 10). *Astronomy Letters*. Retrieved from Springer Link: <http://link.springer.com/search?sortOrder=newestFirst&facet-content-type=Article&facet-journal-id=11443>
- Berriman, B. (2015, May 1). *Approaches to Technology Evaluation in the Era of Data Intensive Astronomy*. Retrieved from Astronomy Computing Today: <https://astrocompute.wordpress.com/2011/05/13/approaches-to-technology-evaluation-in-the-era-of-data-intensive-astronomy/>
- Borne, K. D. (2014, November 18). *Astroinformatics: data-oriented astronomy research and education*. Retrieved from Springer Link: http://rc4ht3qs8p.search.serialssolutions.com.libaccess.hud.ac.uk/?ctx_ver=Z39.88-2004&ctx_enc=info%3Aofi%2Fenc%3AUTF-8&rft_id=info:sid/summon.serialssolutions.com&rft_val_fmt=info:ofi/fmt:kev:mtx:journal&rft.genre=article&rft.atitle=Astroinformatics%3A+D
- Content-Based Image Retrieval in Astronomy. (2014, November 17). *Content-Based Image Retrieval in Astronomy*. Retrieved from ACM Digital Library: <http://dl.acm.org/citation.cfm?id=594007>
- Experimental Astronomy. (2014, December 10). *Springer Link*. Retrieved from Experimental Astronomy: an international journal on astronomical instrumentation and data analysis: <http://link.springer.com.libaccess.hud.ac.uk/search?sortOrder=newestFirst&facet-content-type=Article&facet-journal-id=10686>

- Google Scholar. (2015, July 22). *Google Scholar*. Retrieved from Google:
https://scholar.google.co.uk/scholar?biw=1600&bih=798&dpr=1&bav=on.2,or.&um=1&ie=UTF-8&lr&q=related:kA_HbqYbtv5GhM:scholar.google.com/
- Guillaume, D. (2015, April 28). *Astronomical Markup Language*. Retrieved from The Physical Markup Language - PML: <http://web.mit.edu/mecheng/pml/standards.htm>
- Hall, S. (2015, May 1). *Calling All Volunteers to Help Digitize Astronomical History*. Retrieved from Universe Today: <http://www.universetoday.com/114398/calling-all-volunteers-to-help-digitize-history/>
- International Virtual Observatory Alliance. (2015, April 30). *XML Schema*. Retrieved from International Virtual Observatory Alliance: <http://www.ivoa.net/xml/>
- IVOA Data Model Working Group. (2015, April 29). *IVOA Documents*. Retrieved from International Virtual Observatory Alliance: <http://www.ivoa.net/documents/latest/CharacterisationDM.html>
- Lopez, M. H. (2014, November 14). *The World's Technological Capacity to Store, Communicate, and Compute Information*. Retrieved from Science Magazine:
<http://www.sciencemag.org/content/332/6025/60>
- Mikulski Archive for Space Telescopes (MAST). (2015, February 10). *MAST Web Services*. Retrieved from Mikulski Archive for Space Telescopes (MAST): http://archive.stsci.edu/vo/mast_services.html
- NASA. (2014, October 21). *XDF: The Extensible Data Format Based on XML Concepts*. Retrieved from NASA: http://nssdc.gsfc.nasa.gov/nssdc_news/june01/xdm.html
- NASA. (2015, April 29). *Cover Pages*. Retrieved from FITSML for Flexible Image Transport System:
<http://xml.coverpages.org/FITSML-200103dtd.txt>
- NASA Science. (2014, October 22). *Hubble Space Telescope*. Retrieved from NASA Science:
http://www.nasa.gov/mission_pages/hubble/science/xdm.html
- Openastronomylog. (2014, November 18). *Openastronomylog*. Retrieved from Openastronomylog:
<http://code.google.com/p/openastronomylog/>
- Pepe, A. (2015, July 22). *How Do Astronomers Share Data?* Retrieved from PLOS:
<http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0104798>
- Santander-Vela, J., Garcia, E., Leon, S., Espigares, V., Ruiz, E. J., Verdes-Montenegro, L., et al. (2012). Building a VO-compliant Radio Astronomical Data Model for Single-dish radio telescopes (RADAMS). *Experimental Astronomy*, Volume 34, Issue 3, pp 623-652.
- Shaya, E. (2014, April 21). *FEP - Format Developer – XDF*. Retrieved from ADC Archive:
<http://nssdc.gsfc.nasa.gov/nost/fep/developer-xdf.html>

- SkyQuery: Federating Astronomy Archives. (2014, November 17). *SkyQuery: Federating Astronomy Archives*. Retrieved from |IEEE Xplore Digital Library:
http://ieeexplore.ieee.org.libaccess.hud.ac.uk/xpls/abs_all.jsp?arnumber=6475925
- Strasbourg astronomical Data Center – SIMBAD. (2015, February 10). *SIMBAD*. Retrieved from Strasbourg astronomical Data Center – SIMBAD: <http://simbad.u-strasbg.fr/simbad/sim-help?Page=sim-url>
- Stuart, C. (2015, May 1). *How Big Data is pushing astronomy and statistics in a new direction*. Retrieved from Statistics Views: <http://www.statisticsviews.com/details/feature/5064151/How-Big-Data-is-pushing-astronomy-and-statistics-in-a-new-direction.html>
- Sum, A. K. (2014, December 2). *Introducing Design Patterns in XML Schemas*. Retrieved from Oracle Technology Network: <http://www.oracle.com/technetwork/java/design-patterns-142138.html>
- The Journal of Astronomical Data (JAD). (2014, December 10). *The Journal of Astronomical Data (JAD)*. Retrieved from The Journal of Astronomical Data (JAD):
<http://www.vub.ac.be/STER/JAD/jad.htm>
- The Mikulski Archive for Space Telescopes (MAST). (2014, November 12). *The Mikulski Archive for Space Telescopes (MAST)*. Retrieved from Space Telescope Science Institute: <https://archive.stsci.edu/>
- UK Astronomy Technology Centre. (2014, December 9). *UK Astronomy Technology Centre*. Retrieved from Science and Technology Facilities Council: <http://www.stfc.ac.uk/1185.aspx>
- VAO. (2015, May 1). *History of the Virtual Astronomical Observatory*. Retrieved from VAO:
<http://virtualobservatory.org/whatis/history.aspx>
- Virtual Solar Observatory. (2014, November 7). *Search for Solar Physics Data Products*. Retrieved from Virtual Solar Observatory: <http://sdac.virtualsolar.org/cgi/search>
- W3c Standards. (2014, November 13). *All Standards and Drafts*. Retrieved from W3c Standards:
<http://www.w3.org/TR/tr-date-stds>
- W3C XML Schema Design Patterns. (2014, December 2). *W3C XML Schema Design Patterns*. Retrieved from Microsoft Developer Network: <http://msdn.microsoft.com/en-us/library/aa468564.aspx>
- Wall, M. (2014, November 17). *Astronomy Overload*. Retrieved from Space.com:
<http://www.space.com/9308-astronomy-overload-scientists-shifting-stargazing-data-mining.html>
- XML Schema Patterns for Common Data Structures. (2014, December 2). *XML Schema Patterns for Common Data Structures*. Retrieved from W3: <http://www.w3.org/2005/07/xml-schema-patterns.html>
- XML Technology - Schema. (2014, November 14). *XML Technology - Schema*. Retrieved from W3C:
<http://www.w3.org/standards/xml/schema>

XSL-FO Rendering Online. (2015, March 15). *XSL-FO Rendering Online*. Retrieved from Attic:
<http://www.renderx.com/~renderx/portal/seex/do-it.cgi>